Two new EM-Methods for Boolean Factor Analysis

Húsek, Dušan
2011

Institute of Computer Science
Academy of Sciences of the Czech Republic

Two new EM-Methods for Boolean Factor Analysis

Húsek Dušan, Frolov Alexander, Polyakov Pavel

Technical report No. 1109

10.3.2011

Pod Vodárenskou věží 2, 182 07 Prague 8, phone: +420 266 053 230, fax: +420 286 585 789,
e-mail:dusan@cs.cas.cz

Institute of Computer Science
Academy of Sciences of the Czech Republic

Two new EM-Methods for Boolean Factor Analysis [1]

Húsek Dus̄an, Frolov Alexander, Polyakov Pavel

Technical report No. 1109

10.3.2011

Abstract:

Methods for the discovery of hidden structures of high-dimensional binary data are one of the most important challenges facing the community of machine learning researchers. There are many approaches in the literature that try to solve this hitherto rather ill-defined task. In the present study, we propose a general generative model of binary data for Boolean factor analysis (BFA) and introduce two new Expectation-Maximization BFA algorithms which maximize the likelihood of a Boolean Factor Analysis solution. To show the maturity of our solutions we propose also an informational measure of Boolean Factor Analysis efficiency. Using the so-called bars problem (BP) benchmark, we compare the efficiencies of the proposed algorithms to that of Dendritic Inhibition neural network, Maximal Causes Analysis, and Boolean Matrix Factorization. These methods seem to be the most efficient in BP. Then we discuss the peculiarities of the two methods we proposed and the three mentioned methods in performing BFA.

Keywords:
neural networks, Statistics, Boolean factor analysis, information gain, expectation maximization, dendritic inhibition, Boolean matrix factorization, bars problem

[1] **Acknowledgement:** This work has been partly elaborated not only in the framework of the institutional project AV0Z10300504,but we have also thank to other the institutions that provided financing of our projects under codenames GACR P202/10/0262 and GACR 205/09/1079. We thank to Pavel Bobrov, who is now with VSB TU Ostrava, for help in preparing the manuscript and useful discussion.

1 Introduction

Factor analysis in general is one of the most efficient methods to reveal and to overcome informational redundancy of high-dimensional signals. Boolean Factor Analysis (BFA) as a special case of factor analysis implies that the components of the original signals, factor loadings and factor scores are binary values. Each binary component of the signal can be interpreted as a representation of the appearance or the non-appearance of some attribute in the pattern. The number of considered attributes is the dimension of the signal space, the appearance of an attribute is encoded as One, and its absence as Zero. The patterns are assumed to be composed of many "objects" in different combinations. We define an object as a collection of highly correlated attributes and suppose that objects are relatively independent of one another. Hence the attributes of different objects are only slightly correlated. In terms of BFA, objects are factors, and the presence or absence of a factor in the pattern is identified by the value of the factor score (One or Zero). Correlations between the attributes constituting each factor can be revealed by statistics over the large data set constituted by patterns that contain each factor many times in different combinations with other factors. The aim of BFA is to detect this hidden structure of the signal space and to form a representation in which these independent objects are presented explicitly. A factor may also be interpreted as a hidden cause resulting in the sets of observations [1, 2]. For example in medical research, a cause is a syndrome and an observation is a symptom [3, 4].

In spite of the fact that binary data representations are typical in many fields, including social science, marketing, zoology, genetics, and medicine, BFA methods have only been rather moderately developed. To overcome this lack we first propose a generative model of binary data, adequate for Boolean factor analysis, which is a generalization of the model proposed in [5]. Then we develop an Expectation-Maximization (EM) method [6, 7] which maximizes the likelihood of a Boolean Factor Analysis solution appropriate to the proposed generative model. We call the method EMBFA and its neural network implementation NNBFA. To be able to evaluate the performance of BFA methods we suggest a general information theoretic measure of BFA efficiency which is the difference of two entropies. The first is the entropy of a data set when its hidden factor structure is ignored and the second is the entropy when the data set factor structure is revealed and taken into account. Thus this difference is the information gain provided by BFA. We show that the gain is sensitive to both the noise in the signals and the errors in the BFA results. This analysis allows us to conclude that information gain is a reliable basis for comparing different BFA methods and for detecting the presence of hidden factor structures in a given data set as well.

The well-known benchmark for learning of objects from complex patterns is the bars problem (BP) introduced by [8]. The BP in various modifications has been considered in many papers (for references see [2]). In this problem, each pattern of the data set is an n by n binary pixel image consisting of several of $L = 2n$ possible (one-pixel wide) horizontal and vertical bars (Fig. 1.1). Pixels constituting a bar take the values 1 and pixels not constituting it take the values 0. For each image, each bar could be chosen with the probability C/L, where C is the mean number of bars mixed in an image. At the point of intersection of a vertical and a horizontal bar, the pixel takes the value 1 (OR mixing rule is used). This Boolean summation of pixels belonging to different bars simulates the occlusion of objects. The task is to recognize all bars as individual objects, exploring a data set containing M images consisting of bar mixtures. In most papers where the BP was used as benchmark, C was set to 2 and $n \leq 8$.

In terms of BFA, bars are factors, each image is a Boolean superposition of factors, and the factor score takes the value 1 or 0 depending on the presence or absence of a bar in the image. Thus, the bars problem is a special case of BFA. It was a challenge for us to test EMBFA using BP benchmark, because it is a demonstrative and well investigated example of complex image analysis [9, 2].

In the experimental part of the paper, we compare our methods with three other methods supposed to be most efficient in solving the bars problem. The first method is the Maximal Causes (MCA_3) method suggested by [2]. MCA_3 is also based on an Expectation-Maximization algorithm but a different generative model of signals is used. The second method, the Dendritic Inhibition neural network (DI), was suggested and studied by [10, 11, 9]. The last method is the fast Boolean Matrix Factorization (BMF) suggested by [12].

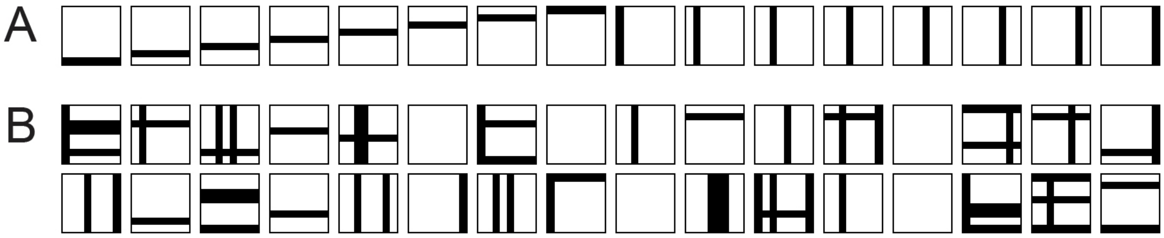


Figure 1.1: **A** Sixteen vertical and horizontal bars in 8 by 8 pixel images. **B** Examples of images in the standard bars problem. Each image contains two bars on average.

This paper is organized as follows. A general generative model is proposed in Section 2. EMBFA, its neural network implementation (NNBFA), and related methods are described in Sections 3, 4, and 5, respectively. In Section 6, we propose the procedure for information gain calculation. The sensitivity of the information gain to signal noise and to inaccuracies in the BFA results is investigated in Section 7. The abilities of the different methods to solve the bars problem are compared in Section 8. The strengths and weaknesses of these BFA methods are discussed in Section 9.

2 A Generative Model of Signals Appropriate for Boolean Factor Analysis

In formulating a generative model of signals appropriate for BFA, we follow ideas of [13, 14], [15], [16], and others who assumed that the search for a hidden factor structure in incoming sensory signals is one of the main brain functions. Explaining Barlow's ideas, [8] writes: "According to [13] objects (and also features, concepts or anything that deserves a name) are collections of highly correlated properties. For instance, the properties 'furry', 'shorter than a meter', 'has a tail', 'moves', 'animal', 'barks', etc. are highly correlated, that is the combination of these properties is much more frequent than it would be if they were independent (the probability of the conjunction is higher than the product of individual probabilities of the component features). It is these non-independent, redundant features, the 'suspicious coincidences' that define objects, features, concepts, categories, and these are what we should be detecting. While components of objects can be highly correlated, objects are relatively independent of one another... The goal of the sensory system might be to detect these redundant features and to form a representation in which these redundancies are reduced and the independent features and objects are represented explicitly."

In terms of BFA, each pattern of a signal space is defined by a binary row vector $\mathbf{x} = [x_1, x_2, \ldots, x_N]$ of dimension N equal to the total number of attributes. Every component of $\mathbf{x}$ takes One or Zero, depending on the presence or absence of the related attribute. Each factor $\mathbf{f}_i = [f_{i1}, f_{i2}, \ldots, f_{iN}]$ is a binary row vector of factor loadings whose One valued entries correspond to highly correlated attributes of the ith object and Zero valued entries correspond to attributes not constituting the object. Although the probability of the object's attribute's appearing in a pattern simultaneously with its other attributes is high, it is not necessarily equal to 1. For example, the attribute "has a tail" does not always appear with the appearance of the object "dog". We denote this probability by p_{ij}, where j is the index of the attribute and i is the index of the factor. For attributes constituting the factor, that is for attributes with $f_{ij} = 1$, the probability p_{ij} is high, and for the other attributes (with $f_{ij} = 0$), it is zero.

As in linear factor analysis, we suppose that in addition to common factors $\mathbf{f}_i$ that influence more than one attribute, each signal also contains N specific or unique factors that influence only particular attributes. Specific factors are also called "specific noise." The contribution of specific factors is defined by a binary row vector $\boldsymbol{\eta} = [\eta_1, \eta_2, \ldots, \eta_N]$ of dimension N. Each specific factor η_j is characterized by the probability q_j with which η_j takes on the value One.

As a result, any observation $\mathbf{x}$ can be presented in the form

$$x_j = [\bigvee_{i=1}^{L} s_i \wedge f'_{ij}] \vee \eta_j, \qquad j = 1, \ldots, N, \tag{2.1}$$

where $\mathbf{s} = [s_1, s_2, \ldots, s_L]$ is a binary row vector of factor scores of dimension L, L being the total number of factors, f'_{ij} is a component of a distorted version $\mathbf{f}'_i = [f'_{i1}, f'_{i2}, \ldots, f'_{iN}]$ of factor $\mathbf{f}_i$ and $\boldsymbol{\eta}$ is a binary row vector of specific factors. Factor distortion implies that One valued entries of $\mathbf{f}_i$ can transform to Zero with probability $1 - p_{ij}$ before mixing in the observed pattern but none of the Zero valued entries $\mathbf{f}_i$ can take One in the distorted version of the factor because the probability for them to transform to One is zero ($p_{ij} = 0$). For ease of presentation we introduce also a short-hand notation of the generative model given by (2.1) similar to the notation of Boolean Matrix Factorization [12]:

$$\mathbf{x} = [\mathbf{s} \otimes \mathbf{F}'] \oplus \boldsymbol{\eta}, \tag{2.2}$$

where $\mathbf{F}'$ is a binary matrix containing distorted factors $\mathbf{f}'_i$ in rows, $\otimes$ is Boolean matrix multiplication, and $\oplus$ is Boolean matrix addition.

We assume that factors appear in patterns (that is related scores s_i take Ones) independently with probabilities π_i ($i = 1, \ldots, L$), factors are distorted independently of other factors and specific factors, factor components are distorted independently of other components, and specific factors are independent of each other and of the common factors.

In principal, without loss of generality, specific factors could be defined as one special common factor $\mathbf{f}_0$ constituting by all attributes and appearing in all patterns (that is $f_{0j} = 1$, $j = 1, \ldots, N$, and $\pi_0 = 1$). Respectively, q_j would be replaced by p_{0j}. Nevertheless, following to notations of linear factor analysis, we prefer to treat this special "common" factor as a set of N specific factors.

The aim of Boolean factor analysis is to find the parameters of a generative model $\boldsymbol{\Theta} = (p_{ij}, q_j, \pi_i, i = 1, \ldots, L, j = 1, \ldots, N)$ and factor scores $\mathbf{s}_m (m = 1, \ldots, M)$ for all M patterns $\mathbf{x}_m$ of the observed data set. However, it is supposed that the factors found could also be detected in any arbitrary pattern if generated by the same model. Note that the finding of p_{ij} implies the finding of factor loadings f_{ij} since $f_{ij} = sgn(p_{ij})$.

The defined generative model has to be discussed in three aspects. First, how, given generative parameters $\boldsymbol{\Theta}$ and scores $\mathbf{s}$, it defines a distribution which assigns a probability to the observation $\mathbf{x}$. This aspect of the model operation corresponds to top-down or feedback procedure since it implies input signal reconstruction by factor scores (Fig. 1(a)). Second, how, given generative parameters $\boldsymbol{\Theta}$ and observation $\mathbf{x}$, the model infers factors contained in the signal, that is, the vector of scores $\mathbf{s}$. This aspect corresponds to bottom-up or feed forward procedure since it implies revealing hidden factors in the input signal or "feature detection" (Fig. 1(b)). Third, how, given a set of observations $\mathcal{X}$, the learning process adjusts the model parameters $\boldsymbol{\Theta}$ and set of scores $\mathcal{S}$ to maximize the probability that the generative model would produce the observed data.

If noise in observations is absent, that is $p_{ij} = f'_{ij} = f_{ij}$ and $q_j = 0$, then, given a vector of factor scores $\mathbf{s}$ and a matrix of factor loadings $\mathbf{F}$ with entries f_{ij}, the observation $\mathbf{x}$ can be easily reconstructed by the formula

$$\mathbf{x} = \mathbf{s} \otimes \mathbf{F}. \tag{2.3}$$

If noise is present, in the top-down procedure only the distribution of $\mathbf{x}$ can be obtained:

$$P(\mathbf{x}|\mathbf{s}, \boldsymbol{\Theta}) = \prod_{j=1}^{N} P(x_j|\mathbf{s}, \boldsymbol{\Theta}) \tag{2.4}$$

where

$$P(x_j|\mathbf{s}, \boldsymbol{\Theta}) = x_j - (2x_j - 1)(1 - q_j) \prod_{i=1}^{L} (1 - p_{ij})^{s_i}. \tag{2.5}$$

To simulate the observation, one must create the matrix $\mathbf{F}'$ containing the common factors distorted according to the p_{ij}, create specific factors according to the q_j, and then mix the common and specific factors according to (2.2). The probability $P(x_j|\mathbf{s},\mathbf{\Theta})$ could be interpreted also as a *prediction* of the observation x_j obtained by soft-OR (or noisy-OR) rule [1].

To reveal the factors contained in a given input signal $\mathbf{x}$ (that is, to perform the bottom-up procedure), a Bayesian approach is the most relevant. By Bayes's theorem,

$$P(\mathbf{s}|\mathbf{x},\mathbf{\Theta}) = \frac{P(\mathbf{x}|\mathbf{s},\mathbf{\Theta})P(\mathbf{s}|\mathbf{\Theta})}{\sum_{\mathbf{s}} P(\mathbf{s}|\mathbf{\Theta})P(\mathbf{x}|\mathbf{s},\mathbf{\Theta})}, \tag{2.6}$$

where

$$P(\mathbf{s}|\mathbf{\Theta}) = \prod_{i=1,L} \pi_i^{s_i}(1-\pi_i)^{1-s_i} \tag{2.7}$$

and $P(\mathbf{x}|\mathbf{s},\mathbf{\Theta})$ is given by (2.4). According to the Bayesian approach, the vector $\mathbf{s}$ which maximizes this probability is taken as the vector of factor scores for $\mathbf{x}$. The global maximum of $P(\mathbf{s}|\mathbf{x},\mathbf{\Theta})$ can be found only by an exhaustive search. However since the number of possible $\mathbf{s}$ is large (equal to 2^L), it is more reasonable to use an iterative procedure providing at least a local maximum. One of the possible procedures is maximizing the likelihood function

$$\mathcal{L}(\mathbf{s}|\mathbf{\Theta}) = \log[P(\mathbf{x}|\mathbf{s},\mathbf{\Theta})P(\mathbf{s}|\mathbf{\Theta})] = \sum_{j=1,N} \log P(x_j|\mathbf{s},\mathbf{\Theta}) + \sum_{j=1,L} \log P(s_i|\mathbf{\Theta}). \tag{2.8}$$

At each iterative step, the values $\mathcal{L}(\mathbf{s}|\mathbf{\Theta})|_{s_i=1}$ and $\mathcal{L}(\mathbf{s}|\mathbf{\Theta})|_{s_i=0}$ obtained by substituting $s_i = 1$ and $s_i = 0$ into $\mathcal{L}(\mathbf{s}|\mathbf{\Theta})$ are compared. The value of s_i that provides the greater $\mathcal{L}(\mathbf{s}|\mathbf{\Theta})|_{s_i}$ is chosen, and the procedure goes to another i until convergence. In our experiments, we used a two-run iterative procedure. At each external cycle of the procedure all the components of $\mathbf{s}$ were processed to maximize $\mathcal{L}(\mathbf{s}|\mathbf{\Theta})$. The sequence of their processing was randomly permuted at each cycle. The procedure was terminated when $\mathbf{s}$ remained the same in the next cycle. The procedure converges because at each iterative step the likelihood function does not decrease. The procedure started with all $s_i = 0$.

3 Expectation-Maximization Method

In this section, we discuss the third aspect of the generative model: searching for its parameters, given a set of input signals $\mathbf{X} = [\mathbf{x}_1,\ldots,\mathbf{x}_M]$. One of the most efficient procedures is the Expectation-Maximization (EM) method [6]. This method allows of finding the parameters of a given probabilistic generative model that maximize the likelihood of the observed data. Since it is applied here to a BFA generative model, we call it EMBFA. The EM method maximizes the likelihood of the observed data by maximizing the free energy [6]. For the proposed generative model it takes the form

$$\mathcal{F} = \sum_{m=1}^{M} \left\{ \sum_{\mathbf{s}} g_m(\mathbf{s}) \big[\log P(\mathbf{x}_m|\mathbf{s},\mathbf{\Theta}) + \log P(\mathbf{s}|\mathbf{\Theta}) \big] + H(g_m(\mathbf{s})) \right\}, \tag{3.1}$$

where $g_m(\mathbf{s})$ is the expected distribution of factor scores for the mth pattern, $H(g_m(\mathbf{s}))$ is the Shannon entropy of $g_m(\mathbf{s})$, $P(\mathbf{x}_m|\mathbf{s},\mathbf{\Theta})$ is given by (2.4), $P(\mathbf{s}|\mathbf{\Theta})$ is given by (2.7), and $\mathbf{\Theta} = (p_{ij}, q_j, \pi_i, i = 1,\ldots,L, j = 1,\ldots,N)$ are the model parameters. The iterations of EM alternatively increase $\mathcal{F}$ with respect to the distributions g_m, while holding $\mathbf{\Theta}$ fixed (the E-step), or with respect to parameters of the model $\mathbf{\Theta}$, while holding g_m fixed (the M-step).

At the E-step, when $\mathbf{\Theta}$ is fixed, the distributions g_m maximizing $\mathcal{F}$ are calculated according to the following equation

$$g_m(\mathbf{s}|\mathbf{\Theta}) = \frac{P(\mathbf{x}_m|\mathbf{s},\mathbf{\Theta})P(\mathbf{s}|\mathbf{\Theta})}{\sum_{\mathbf{s}} P(\mathbf{x}_m|\mathbf{s},\mathbf{\Theta})P(\mathbf{s}|\mathbf{\Theta})}. \tag{3.2}$$

The obtained distributions g_m provide the expected likelihood of the observed data over the factor scores for the given parameters of the generative model [7].

At the M-step, when the distributions g_m are fixed, π_i can be obtained as

$$\pi_i = (1/M) \sum_{m=1}^{M} s_{mi},$$

where

$$s_{mi} = \sum_{\mathbf{s}} g_m(\mathbf{s}|\mathbf{\Theta}) s_i. \tag{3.3}$$

Respectively, p_{ij} and q_j can be obtained by steepest ascent maximization of $\mathcal{F}$:

$$\Delta p_{ij} = \gamma \frac{\partial \mathcal{F}}{\partial p_{ij}}, \qquad \Delta q_j = \gamma \frac{\partial \mathcal{F}}{\partial q_j}, \tag{3.4}$$

where γ is a learning rate,

$$\begin{aligned} \frac{\partial \mathcal{F}}{\partial p_{ij}} &= \sum_{m=1}^{M} \sum_{\mathbf{s}} g_m(\mathbf{s}|\mathbf{\Theta}) P(x_{mj}|\mathbf{s},\mathbf{\Theta})^{-1} \frac{\partial P(x_{mj}|\mathbf{s},\mathbf{\Theta})}{\partial p_{ij}} \\ \frac{\partial \mathcal{F}}{\partial q_j} &= \sum_{m=1}^{M} \sum_{\mathbf{s}} g_m(\mathbf{s}|\mathbf{\Theta}) P(x_{mj}|\mathbf{s},\mathbf{\Theta})^{-1} \frac{\partial P(x_{mj}|\mathbf{s},\mathbf{\Theta})}{\partial q_j} \end{aligned} \tag{3.5}$$

and according to (2.5)

$$\begin{aligned} \frac{\partial P(x_{mj}|\mathbf{s}_m,\mathbf{\Theta})}{\partial p_{ij}} &= (x_{mj} - P(x_{mj}|\mathbf{s}_m,\mathbf{\Theta})) \frac{s_{mi}}{1-p_{ij}} \\ \frac{\partial P(x_{mj}|\mathbf{s}_m,\mathbf{\Theta})}{\partial q_j} &= (x_{mj} - P(x_{mj}|\mathbf{s}_m,\mathbf{\Theta})) \frac{1}{1-q_j}. \end{aligned} \tag{3.6}$$

As we assume that the probabilities p_{ij} are sufficiently high for the components constituting the ith factor ($f_{ij} = 1$) and equal to zero for the other components ($f_{ij} = 0$), at each iteration cycle of step M we put $p_{ij} = 0$ if

$$p_{ij} < 1 - \prod_{l \neq i} (1 - \pi_l p_{lj}), \tag{3.7}$$

where the right side of the inequality is the probability that the jth attribute appears in the pattern due to other factors besides $\mathbf{f}_i$. It is interesting to note that without threshold truncation of p_{ij}, EM does not converge because of uncertainties arising from the competition between factors defined by p_{ij} and noise defined by q_j. For example one of the common factors can contain all Ones, then it will be functionally indistinguishable from specific factors because there is no preference to prescribe Ones of the observed signals to this common factor or to specific factors. To eliminate this uncertainty most of the components of each factor are set to be zero.

The iterative procedure (3.4) at each step M continues until $\sum_{ij} |\Delta p_{ij}|/LN$ become smaller than $\epsilon_2 = 10^{-3}$.

The obtained values of p_{ij}, q_j and π_i are used as the input for the next E–step. EM iterative procedure terminates once values $\sum_{ij} |\Delta p_{ij}|/LN$ remained smaller than $\epsilon_1 = 10^{-3}$.

After the convergence of the procedure, the resulting values s_{mi} are the estimates of the factor scores which are not binary but gradual. To satisfy the generative model, we binarized those values. The binarization threshold was chosen to maximize the BFA information gain described in Section 6.

Here we restricted the EMBFA algorithm to the case of sparse scores, when only a small number of factors (no more than three) are supposed to be mixed in the observed patterns. In this case, summation over $\mathbf{s}$ in the above formulas (3.1, 3.2, 3.3, 3.5) is reduced to

$$\sum_{\mathbf{s}} (\dots) = (\dots)_{\mathbf{s}=0} + \sum_{i} (\dots)_{\mathbf{s}=\mathbf{s}_i} + \sum_{i<j} (\dots)_{\mathbf{s}=\mathbf{s}_{ij}} + \sum_{i<j<k} (\dots)_{\mathbf{s}=\mathbf{s}_{ijk}}, \tag{3.8}$$

where $\mathbf{s}_i$ is the vector of factor scores with all zeros except s_i, $\mathbf{s}_{ij}$ is the vector of factor scores with all zeros except s_i and s_j, and $\mathbf{s}_{ijk}$ is the vector of factor scores with all zeros except s_i, s_j and s_k. An increase of the number of terms in (3.8) leads to a considerable rise in computational complexity.

To start the EM procedure, we set $\pi_i = 1/L$, where L is the expected number of factors; we also initialized $q_j = 0$ and p_{ij} with random values uniformly distributed in the range from 0.2 to 0.7.

EMBFA algorithm is given in Appenix.

4 Neural Network Implementation

Viewed as a neural network (NN), the generative model consists of a layer of N input units whose state vector $\mathbf{x}$ represents the observed binary signal and an output layer of L units whose state vector $\mathbf{s}$ represents factor scores (Fig. 4.1).

During the top-down procedure (Fig. 1(a)) observation $\mathbf{x}$ is reconstructed by the activity $\mathbf{s}$ of output units. The weight of synaptic connection between the jth input unit and the ith output unit amounts to p_{ij} and, hence, the excitation of the input unit by this connection amounts to $p_{ij}s_i$. The excitation of each synapse is supposed to be transmitted to the input unit with probability $p_{ij}s_i$ independently of the excitations of the other synapses. Thus, activation of the ith output unit (when s_i is set to be One) provokes the excitation of the input layer with the binary vector $\mathbf{f}'_i$ which has a Bernoulli distribution

$$P(\mathbf{f}'_i) = \prod_j {p_{ij}}^{f'_{ij}}(1 - p_{ij})^{1-f'_{ij}}$$

According to the generative model, $\mathbf{f}'_i$ is a distorted version of $\mathbf{f}_i$. If distortion of factors is absent ($p_{ij} = f'_{ij} = f_{ij}$), then the activation of ith output unit provokes at the input layer the activation of the ith factor itself.

Each input unit also obtains a binary random excitation from an external source as specific noise. The external excitation of jth input unit η_j takes One with probability q_j. Thus, the vector of specific noise $\boldsymbol{\eta}$ has a Bernoulli distribution

$$P(\boldsymbol{\eta}) = \prod_j {q_j}^{\eta_j}(1 - q_j)^{1-\eta_j}$$

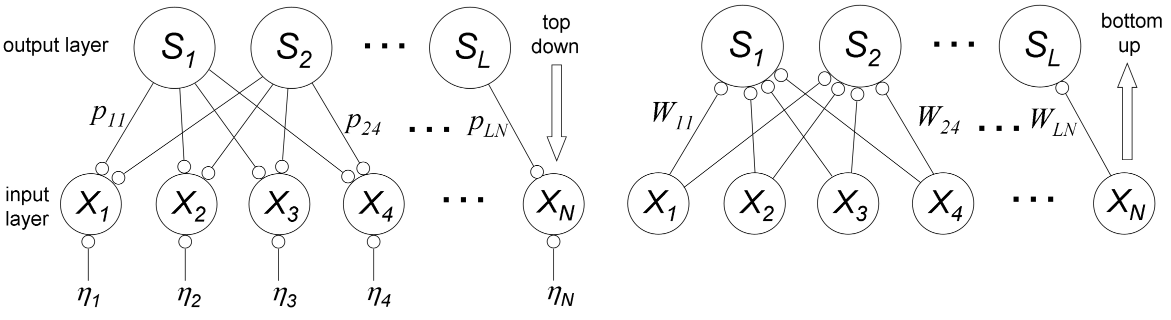
The total excitation transmitted to the jth input unit due to activation of vector $\mathbf{s}$ at the output layer amounts to

$$A_j = \sum_{i=1,L} f'_{ij}s_i + \eta_j$$

and the binary input unit is supposed to be activated if $A_j > 0$ (that is $x_j = sgn(A_j)$). Thus, the jth component of the reconstructed signal $\mathbf{x}$ takes One if the corresponding input unit is activated by any of distorted common factors or a specific factor. The obtained $\mathbf{x}$ is just the same as postulated by the generative model (2.2) and respectively its distribution is given by (2.4). The expectation of $\mathbf{x}$ could be interpreted as a prediction vector $\mathbf{r}$ [1] computed by the soft-OR rule: $r_j = 1-(1-q_j)\prod_{i=1}^{L}(1-s_ip_{ij})$.

During the bottom-up procedure (Fig. 1(b)), the input signal $\mathbf{x}$ is given and each output unit has to be activated only when $\mathbf{x}$ contains the corresponding factor. Thus, the output units are "feature detectors" [14]. Following the Bayesian approach, the state of ith output unit has to be chosen to maximize the likelihood function defined by (2.8). According to (2.5) and (2.8)

$$\begin{aligned}
\mathcal{L}(\mathbf{s}|\boldsymbol{\Theta})|_{s_i=1} &= \sum_{j=1,N}\{x_j\log[1-(1-P_{ij})(1-p_{ij})] + (1-x_j)\log[(1-P_{ij})(1-p_{ij})\}\\
&+ \sum_{l\neq i}^{L}\log[\pi_l(1-\pi_l)] + \log\pi_i\\
\mathcal{L}(\mathbf{s}|\boldsymbol{\Theta})|_{s_i=0} &= \sum_{j=1,N}\{x_j\log P_{ij}) + (1-x_j)\log(1-P_{ij})\} \qquad (4.1)\\
&+ \sum_{l\neq i}^{L}\log[\pi_l(1-\pi_l)] + \log(1-\pi_i),
\end{aligned}$$



(a)

(b)

Figure 4.1: Neural Network implementation of the generative model. $\mathbf{x}$ is the observation activated at the input layer, $\mathbf{s}$ is the vector of factor scores activated at the output (hidden) layer, p_{ij} are the top-down synaptic connections from output to input units, W_{ij} are the bottom-up synaptic connections from input to output units. During the top-down procedure input units are activated by synaptic excitation from the output layer and directly by binary specific factors $\eta_1, \dots, \eta_N$. Each specific factor η_j takes One with probability q_j.

where

$$P_{ij} = 1 - (1 - q_j) \prod_{l \neq i} (1 - p_{lj})^{s_l} \tag{4.2}$$

is the probability that the jth input unit is activated by specific noise or other factors except $\mathbf{f}_i$. Thus to maximize $\mathcal{L}(\mathbf{s}|\mathbf{\Theta})$, the ith output unit is activated if

$$\Delta\mathcal{L} = \mathcal{L}(\mathbf{s}|\mathbf{\Theta})|_{s_i=1} - \mathcal{L}(\mathbf{s}|\mathbf{\Theta})|_{s_i=0} = \sum_{j=1,N} W_{ij} x_j - T_i > 0, \tag{4.3}$$

where

$$W_{ij} = \log \frac{1 + [(1 - P_{ij})/P_{ij}] p_{ij}}{1 - p_{ij}}, \quad T_i = \sum_{j=1,N} \log(1 - p_{ij}) + \log \frac{1 - \pi_i}{\pi_i}.$$

and is not activated in the opposite case.

This formula can be interpreted in terms of the activation of the ith output unit by the input signal $\mathbf{x}$ with threshold T_i and bottom-up synaptic weights W_{ij}. The $W_{ij} = 0$ if $p_{ij} = 0$ and it monotonically increases when p_{ij} increases. Thus, as one could expect, the higher is p_{ij}, the more efficient is the corresponding synaptic connection. However, its efficiency depends not only on p_{ij} but also on probability P_{ij} that the input unit is activated by other factors or specific noise. And again, it is reasonable that W_{ij} increases when P_{ij} decreases because the low probability of an input unit activation by other factors suggests a high confidence of its activation by the ith factor. Since P_{ij} increases when the activity of the other output units increases, the decrease of W_{ij} due to this activity can be considered as a specific kind of a lateral inhibition in the output layer. Since this inhibition affects separately each synaptic connection, it can be called "dendritic inhibition" as suggested by [10].

To implement EMBFA as an NN procedure one has to apply it to each individual signal $\mathbf{x}_m$ instead of the whole learning set $\mathcal{X}$, and hence to maximize

$$\mathcal{L}_m = \sum_{j=1,N} \log P(x_{mj}|\mathbf{\Theta}, \mathbf{s}) \tag{4.4}$$

for each individual signal $\mathbf{x}_m$ instead of the total function $\mathcal{F}$ defined for the whole set $\mathcal{X}$. Since according to (3.6)

$$\frac{\partial \mathcal{L}_m}{\partial p_{ij}} \quad = \quad P(x_{mj}|\mathbf{s}_m, \mathbf{\Theta})^{-1} \frac{\partial P(x_{mj}|\mathbf{s}_m, \mathbf{\Theta})}{\partial p_{ij}} \propto (x_{mj} - P(x_{mj}|\mathbf{s}_m, \mathbf{\Theta})) s_{mi} \tag{4.5}$$

$$\frac{\partial \mathcal{L}_m}{\partial q_j} = P(x_{mj}|\mathbf{s}_m, \mathbf{\Theta})^{-1} \frac{\partial P(x_{mj}|\mathbf{s}_m, \mathbf{\Theta})}{\partial q_j} \propto (x_{mj} - P(x_{mj}|\mathbf{s}_m, \mathbf{\Theta}))$$

then in accordance with (3.6) to provide the increase of $\mathcal{L}_m$ one can put

$$\Delta p_{ij}^m = \gamma(x_{mj} - P(x_{mj}|\mathbf{s}_m, \mathbf{\Theta}))s_{mi} \qquad \Delta q_j^m = \gamma(x_{mj} - P(x_{mj}|\mathbf{s}_m, \mathbf{\Theta})) \tag{4.6}$$

where $P(x_{mj}|\mathbf{s}_m, \mathbf{\Theta})$ are given by (2.5).

As a result, the whole NN procedure for the Boolean factor analysis of signals satisfying the proposed generative model (we call it NNBFA) is the following. Signals of the learning set are sequentially presented to the input layer. For each signal $\mathbf{x}_m$, the activity $\mathbf{s}_m$ at the output layer is calculated according to the bottom-up procedure (4.3) and the model parameters are changed by the rule (4.6). Then, the next input signal is processed, until the learning set is exhausted. Thereafter, signals of the learning set are presented again until the procedure converges according to the same criterion of convergence as used for EMBFA. The values π_i required for the bottom-up procedure are estimated as the frequencies of factor appearance at the previous presentations of the learning set.

To start the NNBFA, we use the same initial distributions for factor scores and for values q_j and p_{ij} as for EMBFA.

Note that in contrast to EMBFA, NNBFA is not restricted to the case of sparse factor scores when only a small number of factors (no more then three) are supposed to be mixed in the observed patterns.

NNBFA algorithm is given in Appenix.

5 Related Methods

In this section, we consider three other BFA related methods. These methods were not developed specially to process data complying with BFA generative model but have been shown to be rather efficient for BFA [12], at least for solving the bars problem [9, 2].

5.1 Boolean Matrix Factorization—BMF

Boolean Matrix Factorization implies a presentation of a binary matrix of observations $\mathbf{X}$ in the form

$$\mathbf{X} = \mathbf{S} \otimes \mathbf{F}, \tag{5.1}$$

where each row of the binary $M \times N$ matrix $\mathbf{X}$ is an observed pattern $\mathbf{x}_m$ $(m = 1, \ldots, M)$, each row of the binary $L \times N$ matrix $\mathbf{F}$ is a representation of a factor $\mathbf{f}_i$ $\quad (i = 1, \ldots, L)$ in the signal space and each row of the binary $M \times L$ matrix $\mathbf{S}$ is a vector of factor scores $\mathbf{s}_m$ $\quad (m = 1, \ldots, M)$ defining which factors are mixed in the pattern $\mathbf{x}_m$. Formula (5.1) completely coincides with (2.2) in the absence of noise when $p_{ij} = f'_{ij} = f_{ij}$ and $q_j = 0$. The method consists in identifying a minimal set of factors that provide a representation of the observed data in the form (5.1). Since this combinatorial problem is NP complete [17] existing methods give reasonable, but not necessarily optimal solutions. Recently [12] revealed a tight relationship between BMF and formal concept analysis [18] and developed two rather simple but efficient algorithms for BMF. In our computer simulations we used the second, faster algorithm. The search for new factors continued until they provide an increase in BFA information gain.

5.2 Dendritic Inhibition Network—DI

This method was developed by [10] for finding parts-based decompositions of images [19, 20]. In general both inputs and outputs of DI are gradual, and thus it can be applied to a much larger class of signals than those complying with BFA generative model. The main idea of the method is to use lateral inhibition of individual synapses instead of total inhibition of a neuron. As a result of network learning, neurons of the output layer acquire specific sensitivity to factors constituting patterns of the data set: the appearance of a factor in the pattern presented to the input layer leads to a strong activation of the related neuron at the output layer. Output neurons can be activated when factors are partially distorted and in the presence of noise. Thus, the activity of each neuron at the output layer provides a gradual estimation of the confidence that this pattern contains the related factor. To transform the gradual activity of the output layer to a binary vector of factor scores we used the same binarization procedure as for EMBFA. Namely we used the binarization threshold maximizing the BFA information gain.

5.3 Expectation-Maximization Method for Maximal Causes Analysis—MCA$_3$

Recently [2] have studied the bars problem with the Expectation-Maximization (EM) method. In the generative model studied by [2], multiple active hidden causes (factors in terms of BFA) combine to determine the values of an observed variable through a max function. Each cause results in a set of observations given by a vector of generative influences (factor loadings in terms of BFA).

If several causes result in the same observation, then the strongest influence alone determines the value of the observed variable. If all influences have the same value, then the max function is equivalent to Boolean summation of the influences and the generative model becomes almost equivalent to the generative model of BFA introduced in Section 2. The difference is in the types of noise used in two generative models. In the generative model introduced in Section 2, noise is supposed to be in the form of factor distortion and specific factors, while [2] supposed noise in the form of a random choice of the observed variable according to a Poisson distribution with mean equal to the strongest influence. Thus it is dealing with integer data and can be applied to a larger class of signals than those complying with BFA generative model. In some sense this model of noise is equivalent to the model of factor distortion proposed here but it ignores noise in the form of specific factors. [2] suggested three EM methods for the described generative model that provided similar results, but the method called MCA$_3$ was slightly better than others. As in EMBFA, the method is restricted to the case of sparse scores when each pattern of the data set contains not more than three factors.

The output of MCA$_3$ are probabilities that patterns of the data set contain found factors. To transform these probabilities to binary factor scores we used the same procedure as described for EMBFA and DI.

6 Information Gain

To evaluate the performance of BFA methods we suggest a general measure of BFA efficiency based on the comparison of two entropies. The first is the entropy of the data set when its hidden factor structure is unknown and the second when it is revealed and taken into account. If the factor structure of the signal space is unknown, then representing the jth component of vector $\mathbf{x}$ requires $h(p_j)$ bits of information, where $h(x) = -x\log_2 x - (1-x)\log_2(1-x)$ is the Shannon function and p_j is the probability of the jth component's taking One. Representing the whole data set requires

$$H_0 = M\sum_{j=1}^{N} h(p_j) \tag{6.1}$$

bits of information. If the hidden factor structure of the signal space is detected and all factor loadings and scores are found, then representing the whole data set requires

$$H = H_1 + H_2 \tag{6.2}$$

bits of information. Here

$$H_1 = M\sum_{i=1}^{L} h(\pi_i) \tag{6.3}$$

is the information required to represent the factor scores and

$$H_2 = \sum_{m=1}^{M}\sum_{j=1}^{N} h(\mathcal{P}(x_{mj}|\mathbf{s}_m, \mathbf{\Theta})) \tag{6.4}$$

where $\mathcal{P}(x_{mj}|\mathbf{s}_m,\mathbf{\Theta})$ is given by (2.5), is the information required to represent all patterns of the data set when factor scores are given. The information gain is determined by the difference between H_0 and H. We define the relative information gain as

$$G = (H_0 - H)/H_0. \tag{6.5}$$

The algorithm is given in Appendix.

According to (6.3) and (6.4), to calculate the information gain one needs to know factor scores s_{mi} ($m = 1,\dots,M$, $i = 1,\dots,L$) for all M patterns of the data set and parameters of the generative model p_{ij} and q_j. However, this information is redundant because the knowledge of factor scores allows the calculation of the generative model parameters and conversely the knowledge of these parameters allows prescribing the proper factor scores to the patterns of the data set. Since most of the methods considered in the sequel (except EMBFA and NNBFA developed in the present paper) do not use the notion of a BFA generative model introduced here, they cannot provide a search for its parameters in principle but do allow of searching for factor scores. That is why our comparison of their efficiencies has to be based on their abilities to assign the proper factor scores to the patterns of the data set. To find the parameters of a BFA generative model required for information gain calculation we suggest a procedure based on maximization of the data set likelihood when the factor scores given by any of the BFA methods considered are fixed.

For the BFA generative model, the data set likelihood function takes the form

$$\mathcal{L}(\mathbf{\Theta}|\mathbf{S}) = \sum_{m=1}^{M} \mathcal{L}_m, \tag{6.6}$$

where $\mathcal{L}_m$ is given by (4.4). One can easily find the maximum of $\mathcal{L}(\mathbf{\Theta}|\mathbf{S})$ by the steepest ascent procedure:

$$\Delta p_{ij} = \gamma \frac{\partial \mathcal{L}(\mathbf{\Theta}|\mathbf{S})}{\partial p_{ij}} \qquad \Delta q_j = \gamma \frac{\partial \mathcal{L}(\mathbf{\Theta}|\mathbf{S})}{\partial q_j}$$

where

$$\begin{aligned} \frac{\partial \mathcal{L}(\mathbf{\Theta}|\mathbf{S})}{\partial p_{ij}} &= \sum_{m=1}^{M} P(x_{mj}|\mathbf{s}_m,\mathbf{\Theta})^{-1} \frac{\partial P(x_{mj}|\mathbf{s}_m,\mathbf{\Theta})}{\partial p_{ij}} \\ \frac{\partial \mathcal{L}(\mathbf{\Theta}|\mathbf{S})}{\partial q_j} &= \sum_{m=1}^{M} P(x_{mj}|\mathbf{s}_m,\mathbf{\Theta})^{-1} \frac{\partial P(x_{mj}|\mathbf{s}_m,\mathbf{\Theta})}{\partial q_j} \end{aligned}$$

and $P(x_{mj}|\mathbf{s}_m,\mathbf{\Theta})$, $\partial P(x_{mj}|\mathbf{s}_m,\mathbf{\Theta})/\partial p_{ij}$ and $\partial P(x_{mj}|\mathbf{s}_m,\mathbf{\Theta})/\partial q_j$ are given by (2.5) and (3.6).

Since we assume that for attributes constituting a factor, the probabilities p_{ij} are sufficiently high, but are equal to zero for other attributes, at each iteration step of this procedure we set $p_{ij} = 0$ according to (3.7).

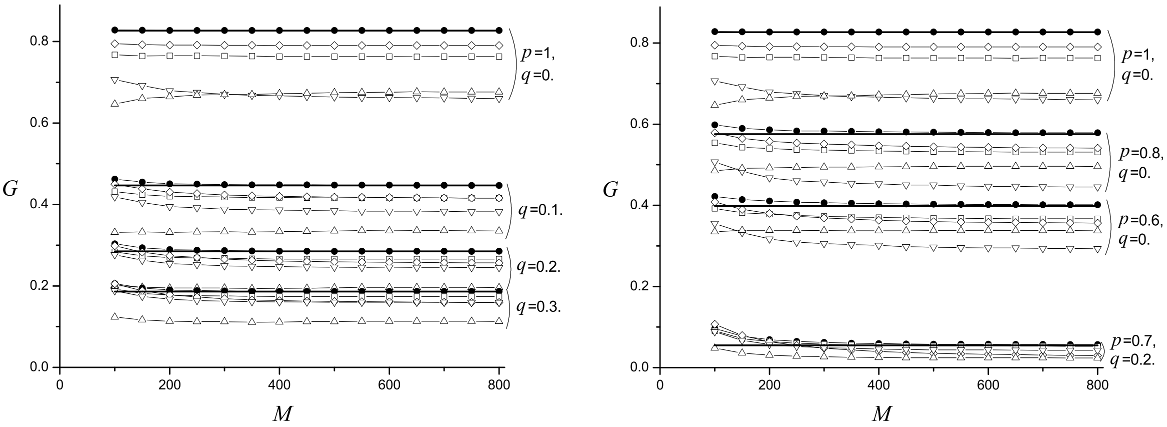
As the input to the steepest ascent procedure, we used p_{ij} and q_j obtained from probabilities p^1_{ij} that the jth attribute appears in the pattern when the ith factor is present and p^0_{ij} when it is absent. On one hand, probabilities p^1_{ij} and p^0_{ij} can be estimated as frequencies of the jth attribute taking One in patterns of the data set containing and not containing the ith factor. On the other hand, these probabilities can be estimated as

$$\begin{aligned} p^0_{ij} &= 1 - (1 - q_j)\prod_{l \neq i}(1 - \pi_l p_{lj}) \\ p^1_{ij} &= 1 - (1 - q_j)(1 - p_{ij})\prod_{l \neq i}(1 - \pi_l p_{lj}) \end{aligned} \tag{6.7}$$

This results in

$$p_{ij} = (p^1_{ij} - p^0_{ij})/(1 - p^0_{ij}).$$

As in the procedure of likelihood maximization, we set the probability p_{ij} equal to zero if it satisfies (3.7). After finding p_{ij}, q_j can be obtained from (6.7). The probabilities π_i are estimated as the



(a) (b)

Figure 7.1: Information gain for "theoretical" (thick lines), "ideal" (thin lines marked by ●) and "erroneous" solutions in dependence on the size of the data set M, □ – one of the factors was excluded, ◇ – 16 false factors in the form of crossing bars were added, ▽ – 10% of randomly chosen scores were excluded, △ – 10% of randomly chosen scores were added. (a) – dependence on q (specific noise) for p = 1, (b) – dependence on p (factors distortion) for q = 0 and for both kinds of noise (q = 0.2, p = 0.7).

frequencies of the related scores provided by BFA. The probabilities p_j required for calculating H_0 are estimated as frequencies of the corresponding components in the data set.

The procedure of the likelihood maximization was always used prior to gain estimation. The algorithm is given in Appendix.

7 Relevance of Information Gain for the Bars Problem

In this section, we illustrate the general properties of the information gain G defined by (6.5), using the bars problem data. Particularly, we compare the values of G for

- the "theoretical solution" when all scores and generative model parameters are exactly the same as those used for data set generation,
- the "ideal solution" when all scores are exactly the same as those used in the generated data set, but the parameters of the generative model are found by data set likelihood maximization (this case simulates the situation when BFA provides scores ideally matching those in the data set under analysis),
- the "erroneous solution" when some factors or scores are missed or false factors or scores are added.

Fig. 7.1 illustrates the dependence of the information gain for "theoretical," "ideal," and "erroneous" solutions on the probabilities q_j and p_{ij}, and on the size of the data set M. Here, $p_{ij} = p$ for components constituting factors ($f_{ij} = 1$) and $q_j = q$ for any j. Recall that $p_{ij} = 0$ for components not constituting factors ($f_{ij} = 0$). In Fig. 7.1, and in subsequent figures, each shown value of G is obtained by averaging over 50 trials. Each trial is based on a data set of a given size M generated according to the model (2.2). The patterns of the data set were 8 by 8 binary images (that is, N = 64). L = 16 vertical and horizontal bars (one pixel width, Fig. 1.1(A)) were randomly mixed in images with probabilities $\pi_i = C/L = 1/8$, thus two bars were mixed in each image on average. Examples of the standard BP images (p = 1, q = 0) are shown in Fig. 1.1(B), and noisy images for p = 0.7, q = 0.2 are shown in Fig. 7.2(A).

For noisy images and small M the information gain for the "ideal" solution is paradoxically higher (Fig. 7.1) than for the "theoretical" one. But this is usually the case for the procedure of likelihood

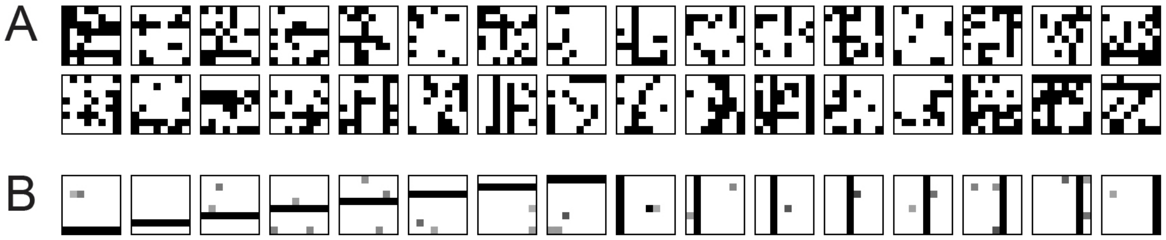


Figure 7.2: **A** Examples of noisy images for $p = 0.7$, $q = 0.2$. **B** Probabilities p_{ij} (shown by the shades of gray) of pixel activation obtained in one of the trials for the ideal solution ($M = 100$, $p = 1$, $q = 0.3$) .

maximization: when the data set is relatively small, the procedure provides the solution for p_{ij} and q_j that better fits randomly obtained peculiarities of a given data set than the "theoretical" solution. That is why both G and $\mathcal{L}(\Theta|\mathcal{S})$ are higher for the "ideal" solution adjusted to those specific peculiarities. Fig. 7.2(B) shows values of p_{ij} obtained for one of the trials with $M = 100$, $p = 1$ and $q = 0.3$. The black pixels correspond to $p_{ij} = 1$, the white pixels correspond to $p_{ij} = 0$, and the gray pixels correspond to the intermediate values. For the pixels constituting bars all $p_{ij} = 1$. However, the factors found by likelihood maximization contain some additional pixels. For some of them, the probability of their appearance with factors is rather high. This means that for this particular data set, those pixels were activated by chance simultaneously with the activation of the related bar, and this peculiarity of the data set was detected by likelihood maximization. When M increases, this effect disappears and the "ideal" solution coincides with the "theoretical" one.

As shown in Fig. 7.1, the maximal information gain is achieved when bars mixed in the images are not distorted ($p = 1$, $q = 0$), and G decreases when noise increases due to both increasing q or decreasing p. We suppose that when the information gain G is positive, BFA is appropriate for a given data set, and when it is negative, BFA makes no sense. The smaller is G, the less explicitly the factor structure of the data set is exposed. For example, when G is small (Fig. 1(b), $p = 0.7$, $q = 0.2$), the bars in the images are almost invisible (Fig. 7.2(A)).

The information gain also decreases when BFA is not perfect. Particularly, G decreases when one of the factors is missing (Fig. 7.1). The decrease of G occurs in this case due to increasing q_j. G also decreases when false factors are added to true factors. In the experiments, to the true 16 factors we added 16 false factors that were crosses of randomly chosen vertical and horizontal bars. As shown below, such false factors are typical for some BFA methods. In the experiment whose results are depicted in Figs. 7.1 and 7.2, the scores for the false factors were given precisely in accordance with their presence in the patterns of the data set. Additional false factors result in decreasing G due to the increase of the first term in (6.2) that gives the information required to describe scores. As shown in Fig. 7.1, G also decreases when true scores were excluded or false scores were added. Thus, all kinds of errors result in decreasing the information gain. Hence, we can conclude that the information gain is a reliable measure for the comparison of different BFA methods and for detecting the presence of a hidden BFA structure in a given data set as well.

8 Performance of BFA Methods in Solving the Bars Problem

In this section, we compare the efficiency of these five methods for Boolean Factor Analysis: BMF, DI, MCA$_3$, EMBFA and NNBFA. These methods are compared according to two criteria. The first one is the information gain introduced in this paper. The second one is a commonly used measure, which is the estimation of the number of true factors L_f^{tr} among the whole set of found factors L_f [9, 2]. Each of the considered BFA methods provides weights evaluating the confidence that the jth attribute pertains to the ith found factor. In BMF, the weights are binary components of a matrix $\mathbf{F}$ in (5.1). In DI, MCA$_3$, EMBFA, and NNBFA, the weights are gradual. In DI, these are the weights of the synaptic connections between neurons of the input and output layers of the network. In MCA$_3$, the weights are represented by influences of causes on the observed variables. In EMBFA and NNBFA, the weights are represented by the probabilities p_{ij}. To estimate L_f^{tr}, the sum of the weights corresponding to each true factor was calculated for each found factor. Since the true factors are horizontal and vertical bars, the sums of the weights for each found factor are calculated over all rows and columns of the image grid. A found factor is considered to represent a particular bar if the sum of the weights corresponding to that bar was twice that of the sum of the weights for any other bar, and if the minimum weight in the row or column corresponding to that bar was greater than the mean of all the weights for that found factor. Note that in contrast to the first criterion, the second one requires a priori knowledge concerning true factors, and thus it can be applied only to artificial data sets when the hidden structure of signals is known in advance. Since this criterion is approved for the bars problem, it is reasonable to compare the performances of the BFA methods by both criteria. Note that for all methods the information gain G is calculated with the use of the likelihood procedure presented in Section 6. In contrast, L_f^{tr} is estimated before this procedure as in the original papers on the bars problem [9, 2].

To binarize the factor scores in EMBFA, DI, and MCA$_3$ we chose the binarization threshold to provide a maximal information gain. Similarly we stopped the search of factors in BMF when the found factors provided a maximal gain.

For EMBFA, NNBFA, DI, and MCA$_3$, the expected number of hidden factors has to be set in advance. In the majority of computer experiments performed by [9] and [2], this number was taken to be twice as great as the actual number of factors. In most cases, such a setup ensured the successful search of all 16 true factors among the 32 factors obtained. In our experiments with EMBFA, NNBFA, DI, and MCA$_3$ the predefined number of hidden factors was also taken to be twice as great as the actual number of factors. In the computer experiments involving DI and MCA$_3$ we used the parameters recommended in the original papers [9, 2]. In EMBFA and NNBFA, the only free parameter is the learning rate γ. We put it to be 0.01 but the results do not depend of its choice within a large range. The advantage of BMF is that it is free of any parameters.

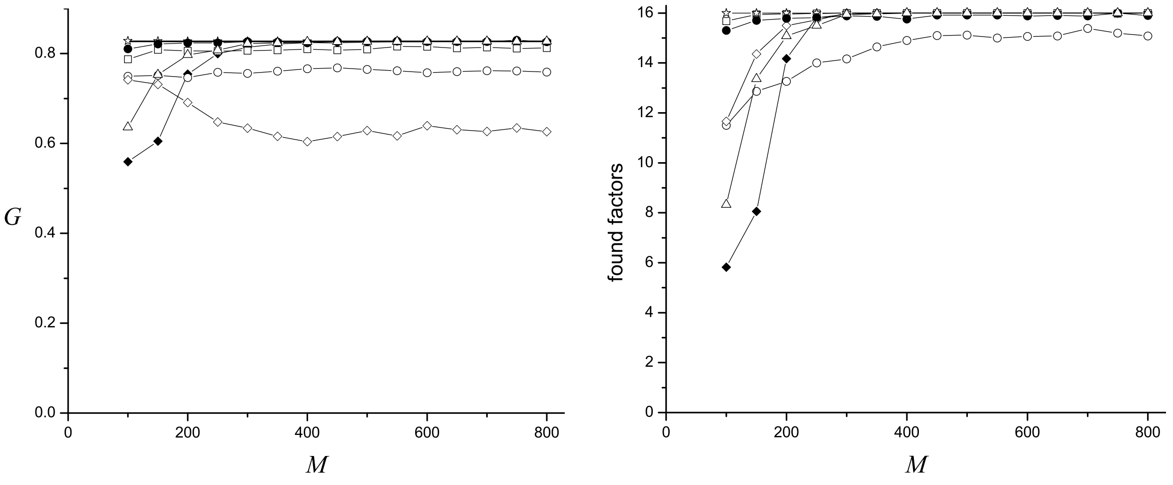
8.1 Noiseless Patterns

Initially, the methods are compared for the case when noise is absent ($p_{ij} = f_{ij}$ and $q_j = 0$).

Standard Bars Problem

We first consider the standard BP described in the previous section when the patterns of the data set are 8 by 8 binary images, the factors are 16 vertical and horizontal bars (one pixel width), and two bars are mixed in each image on average.

As shown in Fig. 8.1, BMF provides an exact solution of the bars problem. Both the information gain and the number of correctly found factors coincide with the ideal solution. When the size of the data set becomes larger ($M \geq 300$), NNBFA also provides an exact solution. For sufficiently large M, DI and MCA$_3$ precisely reveal all true factors (Fig. 1(b)). In spite of the fact that all true factors were found, the information gains obtained by both methods are less than the ideal one. The reason for the decrease in G is the omission of some factor scores. For DI, the fraction of missing scores was 2.3%, and for MCA$_3$ it was to 23%. EMBFA loses to MCA$_3$ in the number of found true factors but wins in information gain. Recall that both EMBFA and MCA$_3$ are able to identify not more than three factors mixed in patterns. For the generative model used, the number of mixed factors k has the binomial distribution $B(k, C/L, L)$, where $C = 2$ and $L = 16$. According to this distribution, 13% of patterns containing more than three factors are generated. Since only three factors can be identified in these pattern, 9% of the scores are expected to be missed. The portion of missed scores for EMBFA amounts to 10% that is close to the theoretical estimation obtained. However for MCA$_3$, this portion is twice as great. The reason is that MCA$_3$ in contrast to EMBFA was sometimes unable to recognize any bar in an image containing a mixture of four or more bars. Some examples of patterns in which three found factors were recognized by EMBFA but not by MCA$_3$ are shown in Fig. 8.2(C). There are two reasons why EMBFA wins. First, in EMBFA false factors are mainly mixtures of two bars (Fig. 8.2(A)) and in MCA$_3$ false factors are mainly duplicates of bars. Then EMBFA is able to identify even five bars in the image as the mixture of three found factors: one is the true factor corresponding to a single bar, and two others are false factors that are the mixtures of two bars.



(a)

(b)

Figure 8.1: Information gain G (a) and number of true found factors L_f^{tr} (b) for five BFA methods in dependence on data set size M. Noise is absent ($q = 0$, $p = 1$). ○ – EMBFA, △ – NNBFA, ☆ – BMF, □ – DI, ◇ – MCA$_3$, ● – EMBFA with fixed number C=2 of mixed bars, ◆ – MCA$_3$ with fixed number C=2 of mixed bars. Thick line – “ideal” solution.

Particularly, the eighth image in Fig. 8.2(C) (images are numbered from left to right) is identified by EMBFA as created by two false factors 1 and 5 in the lower row in Fig. 8.2(A) and the true factor 14 in the upper row in Fig. 8.2(A). The images where four bars were identified by EMBFA as the superposition of three found factors are the first and the last ones in Fig. 8.2(C). The first image is recognized as created by true factors 1 and 8 in the upper row, and by false factor 12 in the lower row in Fig. 8.2(A), and the last one as created by factors 5 and 14 in the upper row and factor 2 in the lower row in Fig. 8.2(A). In other images shown in Fig. 8.2(C), EMBFA could identify three found factors and refer other bars to specific noise. In the generative model for MCA$_3$ specific noise is absent. That is why it cannot identify some bars as factors and other as noise. This is the second and main reason why MCA$_3$ misses factors in complex images while EMBFA identifies them.

To expose the effect of the limitation of both EM algorithms to the case when each image of the data set contains not more than three bars we tested them for the case when each pattern of the data set contains exactly two randomly chosen bars. Those results are shown in Fig. 8.1. For this

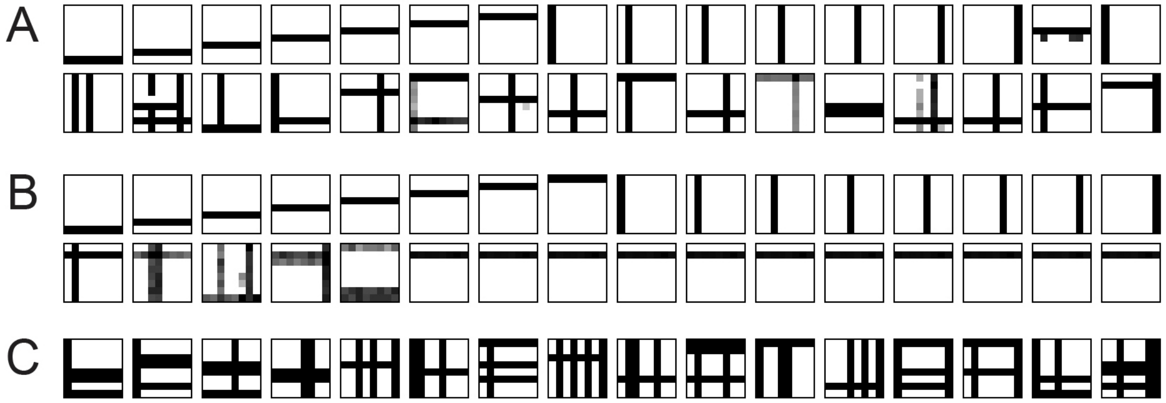


Figure 8.2: Factors found by EMBFA (A) and by MCA$_3$ (B), C – examples of images where bars were identified by EMBFA but not by MCA$_3$. $M = 800$.

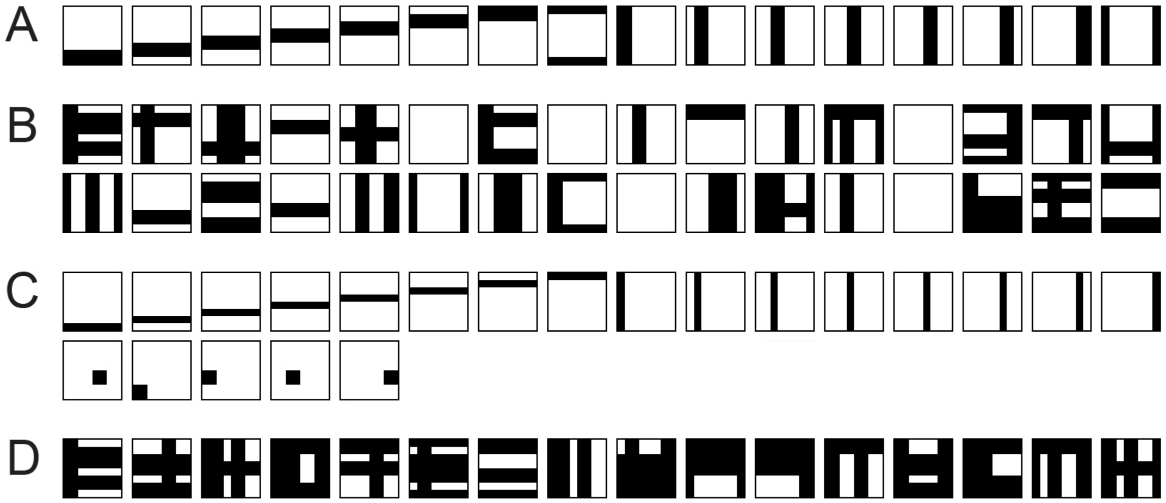


Figure 8.3: **A** Sixteen vertical and horizontal double bars. **B** Examples of images in double bars problem. Each image contains two bars on average. **C** Factors found by BMF in double bars problem. **D** Examples of patterns in double bars problem where DI is unable to reveal any factors.

generative model, EM methods provide a perfect solution. So, we conclude that the observed decrease of information gain in EMBFA and MCA$_3$ compared to the "ideal" one actually results from this restriction.

Special Bars Problems

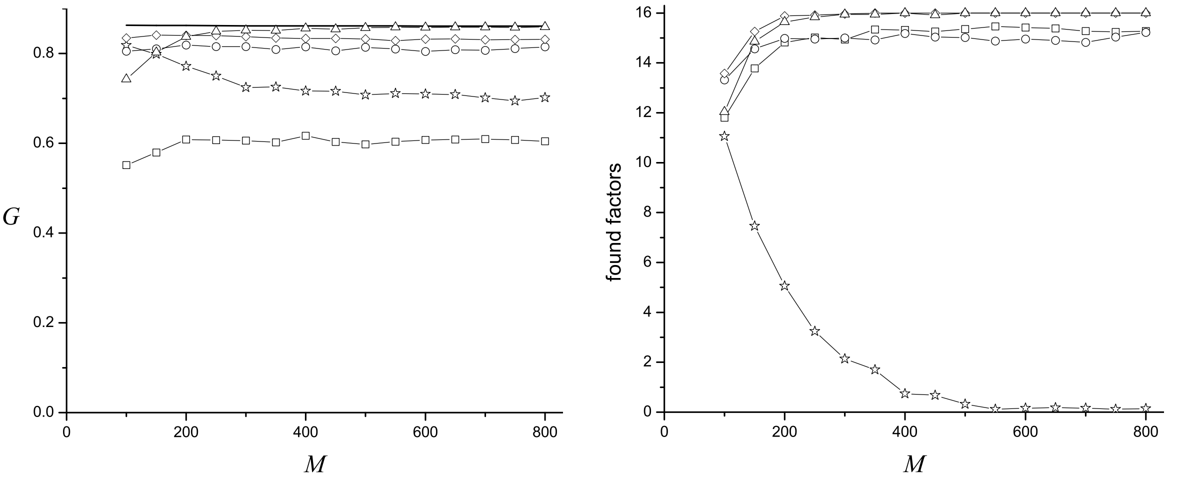
Here we consider two special BP versions. First, we consider strongly overlapping bars when each factor consists of two adjacent bars as shown in Fig. 8.3(A). In this case each of 16 factors is a double bar overlapping with two other factors by half of its pixels. As for the case of single pixel bars, the probability of a factor's appearance in the data set is $\pi_i = 1/8$ for all factors (Fig. 8.3(B)). Second, we consider the case when the factors are standard single pixel bars but they are distributed in the data set with different probabilities π_i.

As shown in Fig. 8.4, NNBFA provides an exact solution for double pixel bars as well as for standard BP. Both the information gain and the number of correctly found factors coincide with the ideal solution. In contrast, BMF loses its high performance in factor finding. The reason is that due to the greedy algorithm used, it finds single pixel bars which are fragments of factors (Fig. 8.3(C)). Nevertheless, BMF provides a rather high information gain because the presentation of the images as superposition of factor fragments instead of as factors also takes into account the hidden structure of the data set but not so accurately.

MCA$_3$ only slightly loses to NNBFA in information gain and EMBFA slightly loses to MCA$_3$ in both the number of true factors found and in information gain. DI loses to the other methods because of the factor score missing. In contrast to the case of single pixel bars, DI found only 62% of the true scores. Examples of data set patterns where DI is unable to reveal any factors are shown in Fig. 8.3(D). All such patterns contain a mixture of three or more double bars. In this case the high activity at the input layer of the DI network suppresses activity at the output layer because of the strong competition between its neurons. Recall that in DI, factor scores are given by the activity of output neurons.

As suggested by [2], to demonstrate the sensitivity of BFA methods to the difference in probabilities of factor appearances in the data set, we consider the case when half of the factors appear in the data set with probability $\pi_i = (1-\alpha)/8$ and the other half with probability $\pi_i = (1+\alpha)/8$. When $\alpha = 0$, the generative model completely coincides with the standard BP considered earlier.

Fig. 8.5 demonstrates the sensitivity of all BFA methods to α. The number of found factors and the information gain are shown for a data set containing $M = 800$ signals. BMF provides an exact



(a) (b)

Figure 8.4: Information gain G (a) and number of true found factors L_f^{tr} (b) in solving double bars problem for five BFA methods in dependence on data set size M. ○ – EMBFA, △ – NNBFA, ☆ – BMF, □ – DI, ◇ – MCA$_3$. Thick line – "ideal" solution.

solution even for $\alpha = 0.9$ when the probabilities of factor appearances differ by 20 times. For all other methods, the number of found factors monotonously decreases when α increases, reaching the value eight for $\alpha = 1$ when actually only eight bars remain to be mixed in patterns of the data set. Although NNBFA for large α slightly loses to BMF in the number of found factors, it provides the same information gain because the lost factors are factors that appear in the data set very rarely and hence do not contribute to its entropy. It is interesting that the gain provided by both EM methods (EMBFA and MCA$_3$) increases when α increases. This results from the fact that in this case the fraction of images containing more than three factors decreases. For example in the limit $\alpha = 1$ (that is when only eight bars instead of 16 are mixed in images but with probabilities double those in the standard BP) the portion of scores expected to be missed by the EM method decreases from the value 0.09 found for the standard BP to 0.07.

8.2 Sensitivity to Noise

Figures 8.6 and 8.8 demonstrate the sensitivity of BFA methods to noise in solving the standard BP. As in Section 7, the noise was assumed to be distributed uniformly over signal components and factors so that $q_j = q$ for any j, and $p_{ij} = pf_{ij}$ for any i and j. We tested BFA methods using data sets of size $M = 800$. As shown in Fig. 8.1, in the absence of noise both the information gain G and the number of found true factors L_f^{tr} reach saturation at that particular M. We checked that saturation is reached for $M = 800$ in the presence of noise also.

Specific Noise: $q > 0$, $p = 1$

As shown in Fig. 6(a) NNBFA is absolutely insensitive to specific noise in terms of the criterion of finding true factors. However its gain is smaller than the "ideal" when $q > 0$. The reason is that it relates several random images which appear more often in the patterns of the data set to factors (Fig. 8.7(A)). In both NNBFA and EMBFA there is a competition between common factors and specific factors. Common factors are defined by probabilities p_{ij}, specific ones by probabilities q_j. To resolve the competition in favor of specific factors the special procedure (3.7) is performed. However for NNBFA it happened to be insufficient and the contribution of specific noise was underestimated and as a result several false common factors looking like random images were found. In contrast, for EMBFA the competition was resolved in favor of specific noise and it performs better than NNBFA.

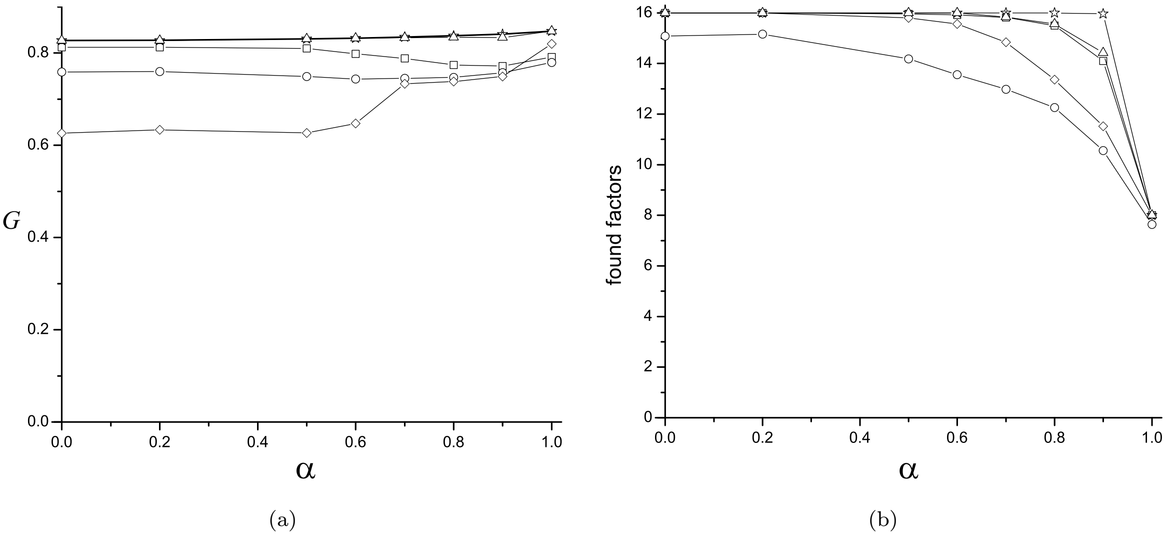
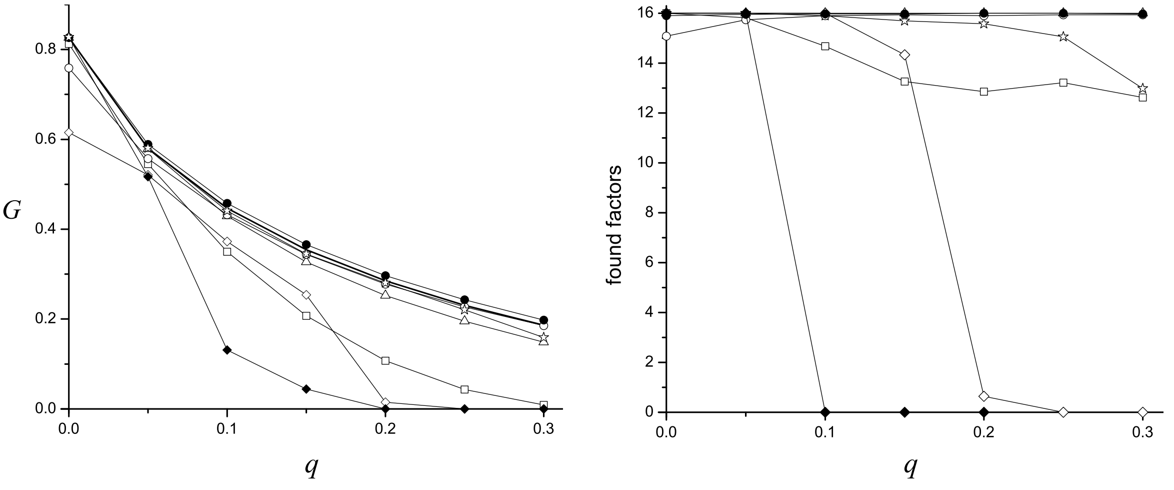


Figure 8.5: Information gain G (a) and number of true found factors L_f^{tr} (b) for five BFA methods in dependence on the parameter α defining the inhomogeneity in distribution of bars in images. ○ – EMBFA, △ – NNBFA, ☆ – BMF, □ – DI, ◇ – MCA$_3$. Thick line – "ideal" solution.

The information gain G obtained by MCA$_3$ and DI demonstrates strong sensitivity to q (Fig. 6(a)). For DI, an increase in q results in a decrease in G, because the number of found true factors decreases (Fig. 6(b)). For MCA$_3$, both G and L_f^{tr} drop near to zero when q increases to 0.2. In this case, the solution of the bars problem by MCA$_3$ becomes unstable, that is it drastically depends on the peculiarities of the data set or on the choice of initial parameters for the EM procedure. With one random realization of the data set MCA$_3$ may provide a perfect solution (after approximately 300 steps of the EM procedure), with another random realization chosen from the same distribution the procedure converges to some random images as factors (in just 3–5 steps). For $q = 0.2$, we observed a successful search for bars by MCA$_3$ only in two of 50 trials.

As mentioned above, the poor performance of MCA$_3$ could be explained by the omission of scores in images containing more than three mixed factors. Then, one could expect that, as for the case without noise (see Fig. 8.1), the information gain should significantly increase if exactly $C = 2$ bars are mixed in every image instead of two on average. However, this does not happen (see Fig. 6(b)) and the above-mentioned instability of MCA$_3$ appears even for smaller q. In particular, a successful search for true factors was not observed in any of the 50 trials already at $q = 0.1$. Therefore the reason for MCA$_3$'s failure is that its generative model is not compatible with the presence of specific noise. This peculiarity of MCA$_3$ was already discussed in Section 8.1.

In contrast, EMBFA performance can be essentially improved by fixing the number of mixed factors in each image of the data set. However, then G paradoxically exceeded the ideal value. In this case EMBFA finds several false factors looking like random images similarly to NNBFA as discussed above (compare Fig. 8.7 A and B). However in this case these images are much more complex and they are identified as factors in patterns of data set much rarer than false factors in NNBFA. In this case the finding of rare false factors is overcompensated by the decrease of specific noise probability q_j. In average it decreases to 0.19, while the exact value used in the generative model is 0.2. It is interesting that the same competition between common and specific factors resulting from the proposed BFA generative model influences oppositely on NNBFA and EMBFA performance. However for both methods, independently of the result of competition, the information gain remains close to the "ideal" one.



(a)

(b)

Figure 8.6: Information gain G (a) and number of found true factors L_f^{tr} (b) in dependence on q for $p = 1$. △ – NNBFA, ○ – EMBFA, ☆ – BMF, □ – DI, ◇ – MCA$_3$, ● – EMBFA with fixed number C=2 of mixed bars, ◆ – MCA$_3$ with fixed number C=2 of mixed bars. Thick line – "ideal" solution.

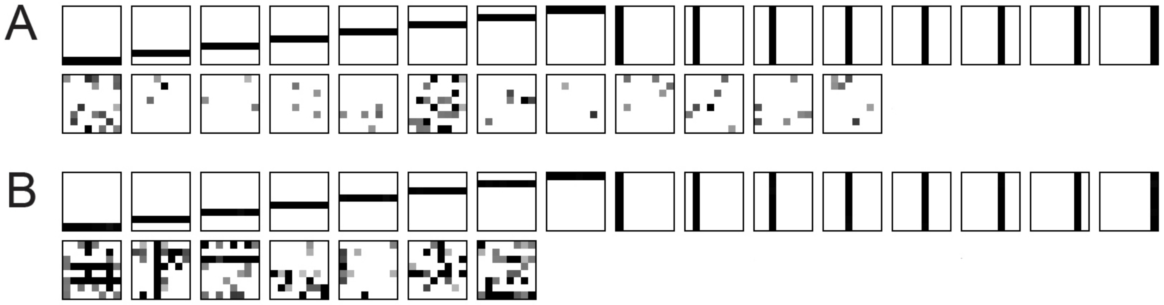


Figure 8.7: Factors found by NNBFA (**A**) and EMBFA (**B**) for $q = 0.3$ and $p = 1$.

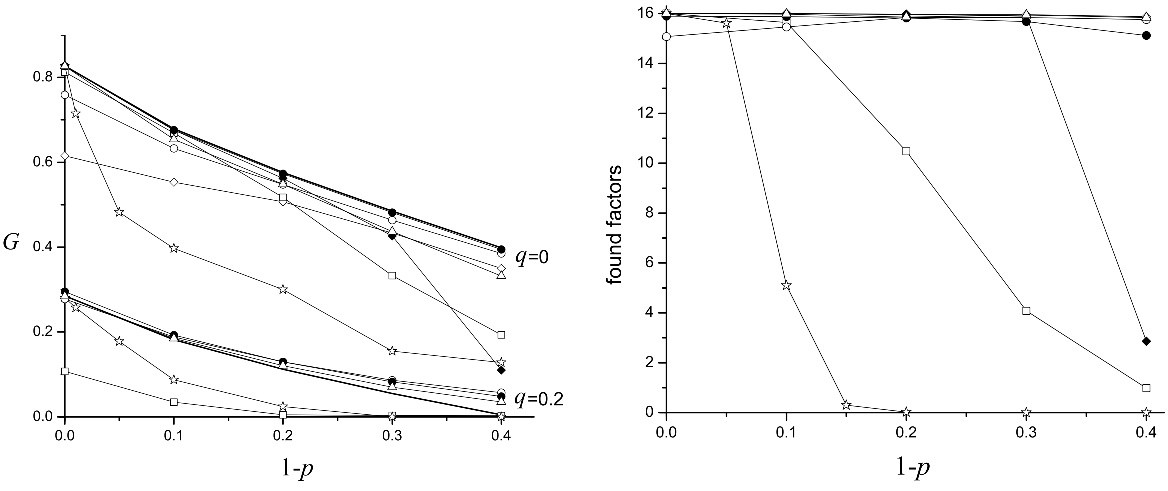
Distortion of Factors: $q = 0$, $p < 1$

BMF happened to be the most sensitive to this kind of noise by both criteria (Fig. 8.8). In this case, factors in the data set are distorted and BMF is able to find only fragments of factors. Thus the number of found true factors decreases. The number of all factors extracted by BMF as a multitude of bar fragments rapidly increases with decreasing p. This results in a gain drop.

DI exhibits a similar sensitivity to factor distortion as to specific noise, while MCA$_3$ is only slightly sensitive to this kind of noise. The reason is that factor distortion is a part of the MCA$_3$ generative model. All true factors are found by this method (Fig. 8(b)) and the information gain G (Fig. 8(a)) is smaller than ideal only due to missing scores in images containing more than three bars, as explained in Section 8.1. Thus, MCA$_3$ performance could be improved by fixing the number of factors in each image. This actually leads to an increase of G. However, when the probability of factor distortion increases (that is p decreases), MCA$_3$ loses stability, as described above for the case of specific noise. For $p = 0.6$, MCA$_3$ provides reasonable solution for only seven out of 50 trials.

EMBFA provides an ideal solution of BP according to both criteria (Fig. 8(a) and Fig. 8(b)). The gain obtained by EMBFA is again higher when the number of bars in each image is fixed, and its gain for this case is even a little higher than the "ideal" one. NNBFA slightly loses to EMBFA because it finds several random images as false factors as described in the previous section.

Fig. 8(a) also demonstrates the sensitivity of BFA methods to both kinds of noise applied simultaneously ($q = 0.2$, $p < 1$). For such noise parameters, MCA$_3$ practically failed (thus, its gain is not



(a)

(b)

Figure 8.8: (a) Information gain G vs. p for $q = 0$ and $q = 0.2$. (b) Number of found factors vs. p for $q = 0$. △ – NNBFA, ○ – EMBFA, ☆ – BMF, □ – DI, ◇ – MCA$_3$, ● – EMBFA with fixed number C=2 of mixed bars, ◆ – MCA$_3$ with fixed number C=2 of mixed bars. Thick line – "ideal" solution.

depicted in Fig. 8(a)). DI provides considerably smaller G than the ideal one. NNBFA and EMBFA perform much better and the information gain G obtained by these methods is again close to the "ideal" one and for small p it is even higher.

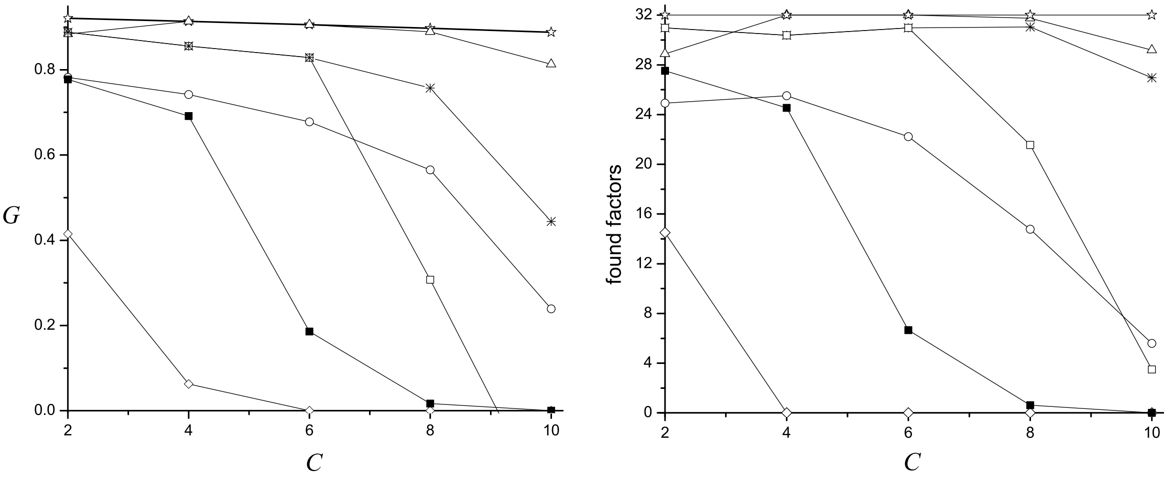
8.3 Sensitivity to the Mean Number of Bars Mixed in Images

To investigate the sensitivity of BFA methods to C, that is, the mean number of bars mixed in images, we increased the size of the images to a grid of 16 by 16 pixels. As in the preceding, we considered two bar-choice cases: 1) each of 32 bars is chosen for each image independently with probability $C/32$ or 2) the number of mixed bars in each image is fixed to C. The increased image size allowed us to study the effects of increasing C up to $C = 10$. Here, we investigated only noiseless case, that is, we put $p_{ij} = f'_{ij} = f_{ij}$, $q_j = 0$.

BMF appeared absolutely insensitive to increasing C, and provided a precise solution of the bars problem even for $C = 10$ (Fig. 8.9). NNBFA only slightly loses to BMF. DI performance gets worse with increasing C, resulting in a decrease in both G and L_f^{tr}. The reason is the loss of solution stability similar to the case for MCA$_3$, described above in Section 8.2. When the number of active neurons at the input of the DI network becomes relatively large because of large C, DI fails. This disadvantage can be overcome by repeating the factor search with other initial synaptic weights. We randomly changed synaptic weights until more than half of the bars were found, but made at most ten attempts. As shown in Fig. 8.9, this modification essentially improved DI performance.

It seems that the presence of images with few mixed factors in the data set facilitates the factor search by DI. To check this hypothesis, we studied DI performance for the case when the number of mixed bars in each image was exactly C (i.e., there were no images containing less than exactly C bars). As shown in Fig. 8.9, this actually significantly worsens DI performance for $C > 4$.

Since MCA$_3$ and EMBFA are both restricted to the case of sparse scores ($C \leq 3$), one would expect that those methods are most sensitive to the increase of C. This actually happened for MCA$_3$: it failed for $C = 4$. However, EMBFA gives reasonable results even until $C = 8$, although with less efficiency. The reason is the above mentioned peculiarity of EMBFA to treat some redundant bars as noise when the number of bars mixed in the image exceeds three.



(a)

(b)

Figure 8.9: Information gain G (a) and number of found factors (b) vs. C for 16 by 16 pixel images at $M = 800$. △ – NNBFA, ○ – EMBFA, ☆ – BMF, □ – DI, ◇ – MCA$_3$, ■ – DI for fixed number of mixed bars in patterns, ✳ – DI with repeated initializations of factor search.

9 Discussion

We now discuss, under three aspects, the results obtained in the study: substantiation of the proposed generative model, validity of the information gain as an index of BFA performance, and comparison of the five investigated BFA methods.

9.1 The Generative Model

The generative model of binary signals suitable for BFA follows the general idea that the external world is organized regularly and the typical form of such regularity is the existence of objects characterized by a set of highly coherent attributes. To create notions of objects, the brain has to calculate the statistics of incoming signals in order to characterize them by some representative variables, the number of which is much smaller than the signal dimension. It is generally accepted that such a reduction of information redundancy of the incoming signals is one of the main functions of the brain [15, 21, 13, 8, 22]. We also believe that this kind of signal redundancy is typical for many fields including social science, marketing, zoology, genetics, medicine and others that operate with nominal data. In the present paper, we consider the application of BFA methods to the bars problem, which is a well-known benchmark test in image analysis [8, 9, 2].

We assume that expression (2.1) provides a rather general form of signal representation, which is a good model defining BFA. Most important here is the introduction of two kinds of noise: the distortion of common factors and a noise in the form of specific factors. The presence of specific factors is a typical assumption of linear factor analysis, whereas distortion of common factors is a peculiarity of the proposed BFA model. For example, for textual data, a factor is some topic characterized by keywords related to factor loadings, and each factor score is defined by whether a given document is dedicated to the topic. Though each topic is represented by a set of keywords, there are no or few documents containing the whole set. Moreover, some keywords are more common for the topic (one can say that they create the topic "kernel") and others are less common (topic "fringe") [23]. Factor distortion means the absence of some keywords from a topic keyword list in a given document dedicated to the topic. Each specific factor relates to each individual word. It is characterized by the probability of the related word to be present in the document independently of topics.

Signals containing certain factors could be grouped. Since a signal can contain several factors, it can be related to several groups. In this aspect, BFA is close to fuzzy clustering. However, BFA provides an explicit knowledge explaining why the signal is shared between clusters. BFA efficiency for fuzzy clustering is demonstrated by [5]. When noise is absent, BFA is equivalent (compare (2.1) and (5.1)) to Boolean matrix factorization [12]. Since in the BFA model, scores are assumed to be nonnegative (1 or 0), it could be related to the methods of Nonnegative Matrix Factorization [24]. Since factor scores are assumed to be sparse (BFA evidently fails for dense factor scores [25]), BFA could be related to the methods of Sparse Component Analysis [26]. Since factors are assumed to be independently distributed in the data set, BFA could be also related to the methods of Independent Component Analysis [27].

9.2 Information Gain

Information gain is the difference between two entropies. The first entropy is calculated for the given signal data set assuming that its factor structure is unknown, and the second one is calculated for that same data set but supposing the factor structure is revealed by BFA and taken into account. The entropy of the data set is obtained as the sum of the Shannon entropies of all the binary signals on the assumption that the signal components are distributed independently. If the data set factor structure is ignored, the probability of each component to take the value One can be evaluated as the frequency of the corresponding Ones in the data set. If the data set factor structure is taken into account, this probability for the jth component and the mth signal is determined by the factors mixed in the signal (that is, by the vector of factor scores $\mathbf{s}_m$ for the signal), and by the probabilities p_{ij} and q_j characterizing factors distortion and specific noise. Since most BFA methods (exceptions are EMBFA and NNBFA) give only factor loadings and factor scores, but not probabilities p_{ij} and q_j, those probabilities have to be evaluated. For their evaluation, we suggest a procedure based on likelihood maximization. This is one of the most powerful and efficient statistical methods that can be easily implemented for the given generative model of signals, and uses as an input from BFA methods only the factor scores, but not the factor loadings.

When the probabilities p_{ij} and q_j are found and the factor scores are given, the probability that the jth component of the mth signal in the data set takes One is given by (2.5). Note that likelihood maximization does not require knowledge of the factor scores distribution. It uses the distribution of factor scores in the given data set provided by BFA.

We have shown that the information gain is sensitive both to noise in the data and to errors in the BFA. When the noise increases (in the form of factor distortion or specific factors), the information gain decreases and becomes zero or negative even if scores are given precisely. Looking at the bar images with small gain (as shown, for example, in Fig. 7.2(A)) one might agree that zero gain corresponds to the threshold when the hidden factor structure in the signals of the data set becomes invisible. Gain also decreases when some true factors are missing or factor scores are found incorrectly. Thus, this gain can be used for comparing different BFA methods. Note that gain is a measure of BFA efficiency that does not require any a priori knowledge concerning signal structure except the assumption that signals satisfy the generative model.

9.3 Performance of the BFA Methods

The bars problem is a common benchmark [9, 2] to reveal strengths and weaknesses of BFA methods. Binary Matrix Factorization (BMF) [12] is perfect in the absence of noise and seems to be insensitive to the size of data set, to the number of mixed factors C in each signal, or to specific noise. However, it loses to other methods in the presence of noise in the form of factor distortion. Oppositely, MCA$_3$ is insensitive to factor distortion but very sensitive to specific noise. Note that the MCA$_3$ generative model is different from the BFA generative model, so its application to BP seems not to be the best test to estimate its general efficiency.

A dendritic inhibition (DI) neural network [9] is moderately sensitive to noise of both kinds but very sensitive to the number of factors mixed in the pattern of the data set. When C increases, the method becomes unstable in the sense that its operation strongly depends on the realization of the initial synaptic weights of the basic network. For one set of initial weights DI may converge to true factors, while for another, it converges to a random solution. This is especially evident for large C (see Fig. 8.9). Another peculiarity of DI is its sensitivity to hints in the data set, that is, to patterns with only a few mixed factors, which seems to significantly facilitate the search for true factors. So, DI fails when the number of mixed factors in each pattern is large and fixed (Fig. 8.9).

The methods EMBFA and NNBFA demonstrate a low sensitivity to noise of both kinds and NNBFA has also a low sensitivity to an increase in C. Better performance of these two methods in solving BP in comparison with DI and MCA$_3$ is expected because they are developed just for the BFA generative model while DI and MCA$_3$ are developed for the other classes of signals. In general, the more assumptions about the data are used in an algorithm, the better can we expect it to perform.

Finally, it is interesting to compare the computational complexity of all the BFA methods considered. Here and below we estimate it in the limit of large M, L, and N. In this limit, the number of operations for BMF is proportional to $\Omega_{BMF} = MLN^2 < n_f >< p_j >$, where $< n_f >$ is the mean number of Ones in the factors and $< p_j >$ is the mean probability of each signal component's being One in the data set. For a PC Core2 6400, 2.13 GHz, the execution time of one operation in seconds amounts to about 10^{-11}. For all methods, this time was estimated by dividing the total execution time by Ω when M, L, and N were sufficiently large so that their doubling resulted in a 5% change in the estimated value.

The number of operations for DI in one iteration step is approximately proportional to $\Omega_{DI} = (2L)MN < p_j >$ and the execution time for one step amounts to about $10^{-7}\Omega_{DI}$. Usually about 15–20 steps are required.

The number of operations required for one iteration step of EMBFA, according to formulas (3.4) and (3.8), is proportional to $\Omega_{EM} = MN(2L)^3 < p_j >$ and the execution time for one step amounts to $5 \cdot 10^{-9}\Omega_{EM}$. For EMBFA the mean number of iteration steps until convergence is about 100. Thus to evaluate the total execution time one must multiply the execution time for one step by a factor of 100.

The number of operations required for one iteration step of MCA$_3$ is the same as for EMBFA. However, the mean time of one operation is approximately twice as high as that for EMBFA, and the mean number of iteration steps until convergence is about 300. Thus, the total execution time for MCA$_3$ is six times higher than for EMBFA.

According to (4.3) and (4.6), the number of operations required for one iteration step of NNBFA is proportional to $\Omega_{NN} = MNL$ and the execution time for one step amounts to $10^{-7}\Omega_{NN}$. Although to start NNBFA as DI, MCA$_3$ and EMBFA the initial number of factors has to be given in advance (as in DI, MCA$_3$ and EMBFA we took it as twice as high as the actual number of factors in the data set), its performance depends only on the actual number of factors because false factors are quickly excluded. For other methods, the given initial number of factors is saved during the BFA performance. That is why in the formulas for Ω_{DI}, Ω_{EM}, and Ω_{MCA_3} we use $2L$ instead of L. Just as for EMBFA, NNBFA usually requires about 100 iteration steps for convergence. Computer simulation revealed that the number of operations in NNBFA also slightly depends on C. For example, Ω_{NN} increases by a factor of 1.5 when C increases from 2 to 10. This increase occurs due to increasing iteration cycles at E step. We ignore this dependence.

For the bars problem $n_f = \sqrt{N}$, $L = 2\sqrt{N}$, and in the absence of noise $< p_j > \simeq C/\sqrt{N}$. As a whole, to perform BFA for a data set of 3200 images of 64 by 64 pixels, containing two undistorted bars, about 460 sec is required for BMF, 110 sec for DI, 240 hours for MCA$_3$, 40 hours for EMBFA, and 2 hours for NNBFA.

In conclusion, we analyzed the strengths and weaknesses of the five common and recently suggested BFA methods. Each of them has a specific application range. BMF is suited and performs well for undistorted data. MCA$_3$ performs well in the absence of specific factors and for a small number of factors in a pattern. DI is only moderately sensitive to both kinds of noise and to the number of factors mixed in the pattern of the data set. The main advantage of DI is its low computational cost. EMBFA and NNBFA are less sensitive to noise and the number of factors in a pattern. When the amount of factors in a data set is large, the computational complexity of EMBFA is higher than that of NNBFA, so in this case NNBFA is preferable. If the properties of a data set are unknown in advance, the best strategy is to perform BFA by all methods, to compare their efficiencies by information gain, and to select the best method for a particular application.

1 Algorithms

Algorithm 1 EMBFA

```
Input:
    $M$: number of observations
    $N$: each observation dimension
    $\{\mathbf{x}_m\}$: set of observations represented as binary vectors of dimension $N$, $m = 1, \ldots, M$
    $L$: expected number of factors
    $Q$: maximum number of factors supposed to be mixed in one object
    $\epsilon_1$, $\epsilon_2$: tolerance parameters used to check convergence
    $\xi$: parameter, $p_{ij}$, $\pi_i$, and $q_j$ are restricted to belong $(\xi; 1 - \xi)$ for the sake of avoiding singularities
Output:
    $\{\mathbf{s}_m\} = \{(s_{m,i})\}$: set of vectors of factor scores expectations, $m = 1, \ldots, M$; $i = 1, \ldots, L$
    $(f_{ij})$: binary factor loadings, $i = 1, \ldots, L$; $j = 1, \ldots, N$
    $\Theta$: set of generative model parameter estimates, $\Theta = \{p_{ij}\} \cup \{\pi_i\} \cup \{q_j\}$, $i = 1, \ldots, L$; $j = 1, \ldots, N$
initialize $p_{ij}$ as random numbers from $[0.2, 0.7]$ for all $i = 1, \ldots, L$; $j = 1, \ldots, N$
initialize $q_j = \xi$ for all $j = 1, \ldots, N$
initialize $\pi_i = 1/L$ for all $i = 1, \ldots, L$
repeat
    $\{g_m(s|\Theta)\}$ = EMBFA_Estep($M, N, \{\mathbf{x}_m\}, L, Q, \Theta$)
    $p_{ij}^{old} = p_{ij}$ for all $i = 1, \ldots, L$; $j = 1, \ldots, N$        ▷ Save old values to check convergence
    $[\{s_m\}, \Theta]$ = EMBFA_MStep($M, N, \{\mathbf{x}_m\}, L, \{g_m(s|\Theta)\}, Q, \Theta, \epsilon_2, \xi$);
until $\sqrt{\sum_{i,j}(p_{ij}^{old} - p_{ij})^2} / \sum_{i,j} p_{ij}^{old} < \epsilon_1$
```

Algorithm 2 EMBFA_EStep

```
Input:
    $M$: number of observations
    $N$: each observation dimension
    $\{\mathbf{x}_m\}$: set of observations represented as binary vectors of dimension $N$, $m = 1, \dots, M$
    $L$: expected number of factors
    $Q$: maximum number of factors supposed to be mixed in one object
    $\Theta$: set of generative model parameter estimates, $\Theta = \{p_{ij}\} \cup \{\pi_i\} \cup \{q_j\}$, $i = 1, \dots, L$; $j = 1, \dots, N$
Output: $\{g(\mathbf{s}|\Theta)\}$: factor score distribution estimates for all observations, $m = 1, \dots, M$
Definition:
    Let $S^v$ be a set of all binary score vectors of dimensionality $L$ which have exactly $v$ non-zero elements
for $v = 1$ to $Q$ do
    for all $\mathbf{s} \in S^v$ do
        compute $P(\mathbf{s}|\Theta) = \prod_{i=1}^{L} \pi_i^{s_i} (1 - \pi_i)^{1-s_i}$
    end for
end for
for $m = 1$ to $M$ do
    $\mathcal{P} = 0$
    for $v = 1$ to $Q$ do
        for all $\mathbf{s} \in S^v$ do
            compute $P(\mathbf{x}_m|\Theta) = \prod_{j=1}^{N} P(x_{m,j}|\mathbf{s}, \Theta)$, where $P(x_{m,j}|\mathbf{s}, \Theta)$ are given by (2.5)
            compute $g_m(\mathbf{s}|\Theta) = P(\mathbf{x}_m|\Theta) P(\mathbf{s}|\Theta)$
            $\mathcal{P} = \mathcal{P} + g_m(\mathbf{s}|\Theta)$    ▷ Accumulate sum of $g_m(\mathbf{s}|\Theta)$ for the fixed value of $m$
        end for
    end for
    divide $g_m(\mathbf{s}|\Theta)$ by $\mathcal{P}$ for all $\mathbf{s}$
end for
```

Algorithm 3 EMBFA_MStep

```
Input:
    $M$: number of observations
    $N$: each observation dimension
    $\{\mathbf{x}_m\}$: set of observations represented as binary vectors of dimension $N$, $m = 1, \ldots, M$
    $L$: expected number of factors
    $Q$: maximum number of factors supposed to be mixed in one object
    $\Theta$: set of generative model parameter estimates, $\Theta = \{p_{ij}\} \cup \{\pi_i\} \cup \{q_j\}$, $i = 1, \ldots, L$; $j = 1, \ldots, N$
    $\{g(\mathbf{s}|\Theta)\}$: factor score distribution estimates for all observations, $m = 1, \ldots, M$
    $\epsilon_1$: tolerance used to check convergence
    $\xi$: parameter, $p_{ij}$, $\pi_i$, and $q_j$ are restricted to belong $(\xi; 1 - \xi)$ for the sake of avoiding singularities
Output:
    $\Theta$: set of generative model parameter estimates, $\Theta = \{p_{ij}\} \cup \{\pi_i\} \cup \{q_j\}$, $i = 1, \ldots, L$; $j = 1, \ldots, N$
    $\{\mathbf{s}_m\} = \{(s_{m,i})\}$: factor score vector expectations, $m = 1, \ldots, M$; $i = 1, \ldots, L$
Definition:
    Let $S^\upsilon$ be a set of all binary score vectors of dimensionality $L$ which have exactly $\upsilon$ non-zero elements
for $m = 1$ to $M$ do
    for $i = 1$ to $L$ do
        $s_{m,i} = \sum_{\upsilon=0}^{Q} \sum_{\mathbf{s} \in S^\upsilon} s_i g(\mathbf{s}|\Theta)$        ▷ Factor score expectation
    end for
end for
for $i = 1$ to $L$ do
    $\pi_i = \frac{1}{M} \sum_{m=1}^{M} s_{m,i}$        ▷ Estimate $\pi_i$
    constraint $\pi_i$ to belong to $[\xi; 1 - \xi]$
end for
```

EMBFA_MStep(continued)

```
repeat
    $p_{ij}^{old} = p_{ij}$ for all $i = 1, \dots, L$; $j = 1, \dots, N$        ▷ Save old values to check convergence
    for $i = 1$ to $L$ do
        for $j = 1$ to $N$ do
            $\frac{\partial \mathcal{F}}{\partial p_{ij}} = \sum_{m=1}^{M} \sum_{v=0}^{Q} \sum_{\mathbf{s} \in S^v} s_i g_m (\mathbf{s}|\Theta) \frac{x_{m,j} - P(x_{m,j}|\mathbf{s},\Theta)}{(1-p_{ij}) P(x_{m,j}|\mathbf{s},\Theta)}$, where $P(x_{m,j}|\mathbf{s},\Theta)$ is given by (2.5)
            $\gamma = p_{ij}(1 - p_{ij})/(M\pi_i)$
            $p_{ij} = p_{ij} + \gamma \frac{\partial \mathcal{F}}{\partial p_{ij}}$
            constraint $p_{ij}$ to belong to $[\xi; 1 - \xi]$
        end for
    end for
    for $j = 1$ to $N$ do
        $\frac{\partial \mathcal{F}}{\partial q_j} = \sum_{m=1}^{M} \sum_{v=0}^{Q} \sum_{\mathbf{s} \in S^v} g_m (\mathbf{s}|\Theta) \frac{x_{m,j} - P(x_{m,j}|\mathbf{s},\Theta)}{(1-q_j) P(x_{m,j}|\mathbf{s},\Theta)}$
        $\gamma = q_j(1 - q_j)/M$
        $q_j = q_j + \gamma \frac{\partial \mathcal{F}}{\partial q_j}$
        constratint $q_j$ to belong to $[\xi; 1 - \xi]$
    end for
    for $i = 1$ to $L$ do
        for $j = 1$ to $N$ do
            if $p_{ij} < 1 - \prod_{l \neq i}(1 - \pi_l p_{lj})$ then
                $p_{ij} = \xi$
            end if
        end for
    end for
until $\sqrt{\sum_{i,j}(p_{ij}^{old} - p_{ij})^2 / \sum_{i,j} p_{ij}^{old}} < \epsilon_1$
```

Algorithm 4 NNBFA

```
Input:
    $M$: number of observations
    $N$: each observation dimension
    $\{\mathbf{x}_m\}$: set of observations represented as binary vectors of dimension $N$, $m = 1, \ldots, M$
    $L$: expected number of factors
    $\gamma$: learining rate
    $\epsilon$: tolerance parameter used to check convergence
    $\xi$: parameter, $p_{ij}$, $\pi_i$, and $q_j$ are restricted to belong $(\xi; 1 - \xi)$ for the sake of avoiding singularities
Output:
    $\tilde{L}$: number of found factors
    $\{\mathbf{s}_m\} = \{(s_{m,i})\}$: set of binary factor score vectors, $m = 1, \ldots, M$; $i = 1, \ldots, \tilde{L}$
    $(f_{ij})$: binary factor loadings, $i = 1, \ldots, L$; $j = 1, \ldots, N$
    $\Theta$: set of generative model parameter estimates, $\Theta = \{p_{ij}\} \cup \{\pi_i\} \cup \{q_j\}$, $i = 1, \ldots, \tilde{L}$; $j = 1, \ldots, N$
    Initialization
    initialize $p_{ij}$ as random numbers from $[0.2, 0.7]$ for all $i = 1, \ldots, L$; $j = 1, \ldots, N$
    initialize $q_j = \xi$ for all $j = 1, \ldots, N$
    initialize $\pi_i = 1/L$ for all $i = 1, \ldots, L$
    $\tilde{L} = L$
    Find the scores and the model parameters
    repeat
        $p_{ij}^{old} = p_{ij}$ for all $i = 1, \ldots, \tilde{L}$; $j = 1, \ldots, N$        ▷ Save old values to check convergence
        for $m = 1$ to $M$ do
            $\mathbf{s}_m$ = NNBFA_LM $(N, \mathbf{x}_m, \tilde{L}, \Theta)$        ▷ Find $\mathbf{s}_m$ by maximizing $\mathcal{L}(\mathbf{s}|\Theta)$
            for $j = 1$ to $N$ do
                compute $P(x_{m,j}|\mathbf{s}_m, \Theta) = x_{m,j} - (2x_{m,j} - 1)(1 - q_j) \prod_{i=1}^{\tilde{L}} (1 - p_{ij})^{s_{m,i}}$
            end for
            for $i = 1$ to $L$ do
                for $j = 1$ to $N$ do
                    $p_{ij} = p_{ij} + \gamma s_{m,i} (x_{m,j} - P(x_{m,j}|\mathbf{s}_m, \Theta))$
                    constraint $p_{ij}$ to belong to $[\xi; 1 - \xi]$
                end for
            end for
            for $j = 1$ to $N$ do
                $q_j = q_j + \gamma (x_{m,j} - P(x_{m,j}|\mathbf{s}_m, \Theta))$
                constraint $q_j$ to belong to $[\xi; 1 - \xi]$
            end for
        end for
```

NNBFA(continued)

```
    for $i = 1$ to $\tilde{L}$ do
        for $j = 1$ to $N$ do
            if $p_{ij} < 1 - \prod_{l \neq i}(1 - \pi_l p_{lj})$ then
                $p_{ij} = \xi$
            end if
        end for
    end for
    $l = 0$
    for $i = 1$ to $\tilde{L}$ do
        $\pi_i = \frac{1}{M} \sum_{m=1}^{M} s_{m,i}$
        if $\pi_i = 0$ OR $\sum_{j=1}^{N} p_{ij} \leq N\xi$ then
            $l = l + 1$
            discard $s_{m,i}$, $\pi_i$, and $p_{ij}$ for all $j = 1, \ldots, N$
        end if
    end for
    $L = L - l$
until $\sqrt{\sum_{i,j}(p_{ij}^{old} - p_{ij})^2 / \sum_{i,j} p_{ij}^{old}} < \epsilon$
Compute factor loadings
for $i = 1$ to $L$ do
    for $j = 1$ to $N$ do
        $f_{ij} = \text{sign}(p_{ij} - \xi)$
    end for
end for
```

Algorithm 5 NNBFA_LM

```
Input:
    $N$: each observation dimensionality
    $\mathbf{x}$: an observation
    $\tilde{L}$: expected number of factors
    $\Theta$: set of generative model parameter estimates, $\Theta = \{p_{ij}\} \cup \{\pi_i\} \cup \{q_j\}$, $i = 1, \ldots, \tilde{L}$; $j = 1, \ldots, N$
Output:
    $\mathbf{s}$: binary vector of factor scores maximizing $\mathcal{L}(\mathbf{s}|\Theta)$
initialize $\mathbf{s}$ as vector of zeros
repeat
    $\mathbf{s}^{old} = \mathbf{s}$
    generate perm, a random permutation from 1 to $\tilde{L}$        ▷ Different at each iteration
    for $k = 1$ to $\tilde{L}$ do
        $i = \mathbf{perm}(k)$        ▷ The $k$-th number of the permutation perm
        for $j = 1$ to $N$ do
            compute $P_{ij} = (1 - q_j) \prod_{l \neq i}(1 - p_{lj})^{s_l}$
            compute $W_{ij} = \log\left(\frac{1 + p_{ij} P_{ij}/(1 - P_{ij})}{1 - p_{ij}}\right)$
        end for
        $T_i = \log \pi_i - \log(1 - \pi_i) + \sum_{j=1}^{N} \log(1 - p_{ij})$
        if $\sum_{j=1}^{N} W_{ij} x_j + T_i > 0$ then
            $s_i = 1$
        else
            $s_i = 0$
        end if
    end for
until $\mathbf{s}^{old}$ is equal to $\mathbf{s}$ ($\mathbf{s}$ is unchanged)
```

Algorithm 6 BFA Gain

```
Input:
    $M$: number of observations
    $N$: each observation dimension
    $\{\mathbf{x}_m\}$: set of observations represented as binary vectors of dimension $N$, $m = 1, \dots, M$
    $L$: number of factors
    $\{\mathbf{s}_m\} = \{(s_{m,i})\}$: set of binary factor score vectors, $m = 1, \dots, M$; $i = 1, \dots, L$
    $\epsilon$: tolerance parameter used to check convergence
    $\xi$: parameter, $p_{ij}$, $\pi_i$, and $q_j$ are restricted to belong $[\xi; 1 - \xi]$ for the sake of avoiding singularities
Output:
    $G$: information gain
$\Theta$ =BFA_LM$(M, N, \{\mathbf{x}_m\}, \{\mathbf{s}_m\}, L, \epsilon, \xi)$
for $j = 1$ to $N$ do
    $p_j = \frac{1}{M} \sum_{m=1}^{M} x_{m,j}$
end for
$H_0 = M \sum_{j=1}^{N} h(p_j) = M \sum_{j=1}^{N} [-p_j \log_2(p_j) - (1 - p_j) \log_2(1 - p_j)]$
$H_1 = M \sum_{i=1}^{L} h(\pi_i)$
$H_2 = \sum_{m=1}^{M} \sum_{j=1}^{N} h\left(P\left(x_{m,j} | \mathbf{s}_m, \Theta\right)\right)$, where $P\left(x_{m,j} | \mathbf{s}, \Theta\right)$ is given by (2.5)
$G = (H_0 - H_1 - H_2)/H_0$                                  ▷ The information gain
```

Algorithm 7 BFA_LM

Input:

M: number of observations

N: each observation dimension

$\{\mathbf{x}_m\}$: set of observations represented as binary vectors of dimension N, $m = 1, \dots, M$

L: number of factors

$\{\mathbf{s}_m\} = \{(s_{m,i})\}$: set of binary factor score vectors, $m = 1, \dots, M$; $i = 1, \dots, L$

ϵ: tolerance parameter used to check convergence

ξ: parameter, p_{ij}, π_i, and q_j are restricted to belong $(\xi; 1 - \xi)$ for the sake of avoiding singularities

Output:

Θ: set of generative model parameter estimates, $\Theta = \{p_{ij}\} \cup \{\pi_i\} \cup \{q_j\}$, $i = 1, \dots, L$; $j = 1, \dots, N$, with p_{ij} and q_j maximizing the likelihood given the scores

```
for $i = 1$ to $L$ do
    $\pi_i = \frac{1}{M} \sum_{m=1}^{M} s_{m,i}$
    constraint $\pi_i$ to belong to $[\xi; 1 - \xi]$
end for
for $i = 1$ to $L$ do
    for $j = 1$ to $N$ do
        $p_j = \frac{1}{M} \sum_{m=1}^{M} x_{m,j}$
        if $\{m : s_{m,i} = 1\} \neq \varnothing$ then
            $p_{ij}^1 = \frac{1}{\#\{m:s_{m,i}=1\}} \sum_{m:s_{m,i}=1} x_{m,j}$
        else
            $p_{ij}^1 = 0$
        end if
        constraint $p_{ij}^1$ to belong to $[\xi; 1 - \xi]$
        if $\{m : s_{m,i} = 0\} \neq \varnothing$ then
            $p_{ij}^0 = \frac{1}{\#\{m:s_{m,i}=0\}} \sum_{m:s_{m,i}=0} x_{m,j}$
        else
            $p_{ij}^0 = 0$
        end if
        constraint $p_{ij}^1$ to belong to $[\xi; 1 - \xi]$
    end for
end for
```

BFA_LM(continued)

```
for $i = 1$ to $L$ do
    for $j = 1$ to $N$ do
        $p_{ij} = \frac{p^1_{ij} - p^0_{ij}}{1 - p^0_{ij}}$
        constraint $p_{ij}$ to belong to $[\xi; 1 - \xi]$
    end for
end for
for $i = 1$ to $L$ do
    for $j = 1$ to $N$ do
        if $p_{ij} < 1 - \prod_{l \neq i}(1 - \pi_l p_{lj})$ then
            $p_{ij} = \xi$
        end if
    end for
end for
for $j = 1$ to $N$ do
    $q_j = 1 - \frac{1 - p_j}{\prod_{l=1}^{L}(1 - \pi_l p_{lj})}$
    constraint $q_j$ to belong to $[\xi; 1 - \xi]$
end for
repeat
    $p^{old}_{ij} = p_{ij}$ for all $i = 1, \ldots, L$; $j = 1, \ldots, N$          ▷ Save old values to check convergence
    for $i = 1$ to $L$ do
        for $j = 1$ to $N$ do
            $\frac{\partial \mathcal{L}(\Theta|S)}{\partial p_{ij}} = \sum_{m=1}^{M} s_{m,i} \frac{x_{m,j} - P(x_{m,j}|\mathbf{s}, \Theta)}{(1 - p_{ij}) P(x_{m,j}|\mathbf{s}, \Theta)}$, where $P(x_{m,j}|\mathbf{s}, \Theta)$ is given by (2.5)
            $\gamma = p_{ij}(1 - p_{ij})/(M\pi_i)$
            $p_{ij} = p_{ij} + \gamma \frac{\partial \mathcal{L}(\Theta|S)}{\partial p_{ij}}$
            constraint $p_{ij}$ to belong to $[\xi; 1 - \xi]$
        end for
    end for
    for $j = 1$ to $N$ do
        $\frac{\partial \mathcal{L}(\Theta|S)}{\partial q_j} = \sum_{m=1}^{M} \frac{x_{m,j} - P(x_{m,j}|\mathbf{s}, \Theta)}{(1 - q_j) P(x_{m,j}|\mathbf{s}, \Theta)}$
        $\gamma = q_j(1 - q_j)/M$
        $q_j = q_j + \gamma \frac{\partial \mathcal{L}(\Theta|S)}{\partial q_j}$
        constraint $q_j$ to belong to $[\xi; 1 - \xi]$
    end for
    for $i = 1$ to $L$ do
        for $j = 1$ to $N$ do
            if $p_{ij} < 1 - \prod_{l \neq i}(1 - \pi_l p_{lj})$ then
                $p_{ij} = \xi$
            end if
        end for
    end for
until $\sqrt{\sum_{i,j}(p^{old}_{ij} - p_{ij})^2 / \sum_{i,j} p^{old}_{ij}} < \epsilon$
```

Bibliography

[1] E. Saund, "A multiple cause mixture model for unsupervised learning," *Neural Computation*, vol. 7, no. 1, pp. 51–71, 1995.

[2] J. Lücke and M. Sahani, "Maximal causes for non-linear component extraction," *The Journal of Machine Learning Research*, vol. 9, pp. 1227–1267, 2008.

[3] A. C. Weber and C. Scharfetter, "The syndrome concept: history and statistical operationalizations." *Psychol Med*, vol. 14, no. 2, pp. 315–325, 1984.

[4] H. O. Veiel, "Psychopathology and Boolean Factor Analysis: a mismatch." *Psychol Med*, vol. 15, no. 3, pp. 623–628, 1985.

[5] A. A. Frolov, D. Husek, and P. Y. Polyakov, "Recurrent neural network based Boolean factor analysis and its application to automatic terms and documents categorization," *IEEE Transactions on Neural Networks*, vol. 20, no. 7, pp. 1073–1086, 2009.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[7] R. M. Neal and G. E. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," *Learning in graphical models*, vol. 89, pp. 355–368, 1998.

[8] P. Foldiak, "Forming sparse representations by local anti-hebbian learning," *Biological Cybernetics*, vol. 64, p. 165170, 1990.

[9] M. W. Spratling, "Learning image components for object recognition," *Journal of Machine Learning Reasearch*, vol. 7, pp. 793–815, 2006.

[10] M. W. Spratling and M. H. Johnson, "Preintegration lateral inhibition enhances unsupervised learning," *Neural Computation*, vol. 14, no. 9, pp. 2157–2179, 2002.

[11] ——, "Exploring the functional significance of dendritic inhibition in cortical pyramidal cells," *Neurocomputing*, vol. 52, no. 54, pp. 389–395, 2003.

[12] R. Belohlavek and V. Vychodil, "Discovery of optimal factors in binary data via a novel method of matrix decomposition," *Journal of Computer and System Sciences*, vol. 76, no. 1, pp. 3–20, 2010.

[13] H. B. Barlow, "Cerebral cortex as model builder," in *Models of the visual cortex*, D. Rose and V. G. Dodson, Eds. Chichester: Wiley, 1985, pp. 37–46.

[14] H. Barlow, "Redundancy reduction revisited," *Network: Computation in Neural Systems*, vol. 12, no. 3, pp. 241–253, 2001.

[15] D. Marr, "A Theory for Cerebral Neocortex," *Proceedings of the Royal Society of London. Series B, Biological Sciences (1934-1990)*, vol. 176, no. 1043, pp. 161–234, 1970.

[16] E. M. Kussul, *Associative neuron-like structures.* Kiev: Naukova Dumka, 1992.

[17] P. Miettinen, T. Mielikainen, A. Gionis, G. Das, and H. Mannila, “The discrete basis problem,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 10, pp. 1348–1362, 2008.

[18] B. Ganter, R. Wille, and R. Wille, *Formal concept analysis.* Springer Berlin, 1999.

[19] P. O. Hoyer, “Non-negative matrix factorization with sparseness constrains,” *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[20] D. Lee and H. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[21] D. Marr, “Simple Memory: A Theory for Archicortex,” *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences (1934-1990)*, vol. 262, no. 841, pp. 23–81, 1971.

[22] K. Doya, “What are the computations of the cerebellum, the basal ganglia and the cerebral cortex?” *Neural networks*, vol. 12, no. 7-8, pp. 961–974, 1999.

[23] V. J. Hodge and J. Austin, “Hierarchical word clustering - automatic thesaurus generation,” *Neurocomputing*, vol. 48, pp. 819–846, 2002.

[24] S. Zafeiriou, A. Tefas, I. Bucie, and I. Pitas, “Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification,” *IEEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 683–695, 2006.

[25] A. A. Frolov, D. Husek, I. P. Muraviev, and P. Y. Polyakov, “Boolean factor analysis by attractor neural network,” *IEEE Transactions on Neural Networks*, vol. 18, no. 3, pp. 698–707, 2007.

[26] P. Georgiev, F. Theis, and A. Cichocki, “Sparse component analysis blind sourse separation of underdetermined mixtures,” *IEEE Transactions on Neural Networks*, vol. 16, no. 4, pp. 992–996, 2005.

[27] Z. Koldovsky, P. Ticharsky, and E. Oja, “Efficient variant of algorithm fast ica for independent component analysis attaining the cramer-rao lower bound,” *IEEE Transactions on Neural Networks*, vol. 17, no. 5, pp. 1265–1277, 2006.